RetroMagazine
Anno 2 - Numero 4

FEBBRAIO 2018 - WWW.RETROMAGAZINE.NET

IN EVIDENZA IN QUESTO NUMERO

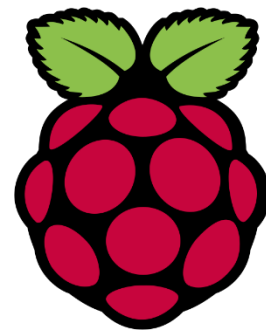
Quando il futuro era ad 8 bit...

di Marco Pistorio

Ha un senso, oggi, parlare di retrocomputing e di "retrogames"? Beh, dal momento che trattiamo tali argomenti e ci leggete ormai in tanti, direi proprio di sì. Da cosa nasce questo desiderio di riscoprire queste tematiche? Probabilmente i motivi sono tanti. Innanzitutto, per molti di noi che abbiamo compiuto i primi passi nel mondo dell'informatica con queste macchine, direi che è un modo per rinfrescarsi la memoria, riprovare quelle emozioni degli anni della nostra infanzia ed adolescenza e quindi, contestualmente, un modo per sentirsi giovani, almeno nel nostro intimo. Qualcuno la chiama la "sindrome di Peter Pan". Ma probabilmente non è questo l'unico aspetto della questione da tenere in considerazione. Per tanti di noi, tali macchine furono oggetto di gioco ma anche di studio, a volte proficuo a volte meno. Negli anni '80 i mezzi a disposizione per scambiarsi opinioni, informazioni tecniche e non, esempi di codice, programmi etc. erano scarsi. Oggi,a differenza di allora, le informazioni sono abbondanti e le fonti sono diverse e di facile accesso per tutti. Ecco quindi che nasce la voglia di comprendere meglio ciò che si comprendeva poco allora o che si è dimenticato strada facendo, nasce la voglia di "rimettersi in gioco", nasce la voglia di tornare a digitare programmi BASIC oppure in Assembly. Ed in questo contesto, anche noi di "RetroMagazine" proveremo a stimolarVi con articoli che vertono (e verteranno) su molteplici aspetti di questo retromondo, statene certi :) Ma se poteva essere utile allora, quando "il futuro era ad 8 bit", conoscere in maniera più approfondita queste macchine e le relative nozioni per programmarle, per realizzare dei videogiochi ad esempio, oggi, nel 2018, può ancora esserlo? Beh, qui è difficile dare una risposta obiettiva. Oggi chi sviluppa software ed hardware mirato per questo settore lo fa più per soddisfazione personale che non per i potenziali guadagni. Tuttavia non è detto che qualcuno di Voi non possa riuscire sia nell'una che nell'altra cosa ovvero, oltre che trarne delle soddisfazioni a titolo personale, magari anche guadagnarci su, perché no? In fondo questo è un mercato di "nicchia" che in qualche (fortunato) caso si riesce a sfruttare. Di certo però, imparare meglio alcuni aspetti relativi a queste macchine, oltre a farci sentire forse un po' più giovani "dentro" ci aiuterà a comprendere l'evoluzione della tecnologia, ci aiuterà forse ad apprezzarla meglio e magari ad intravederne i futuri sviluppi. Vi lasciamo ora alla lettura di questo nuovo numero 4 di RetroMagazine!

Programmazione Atari 2600

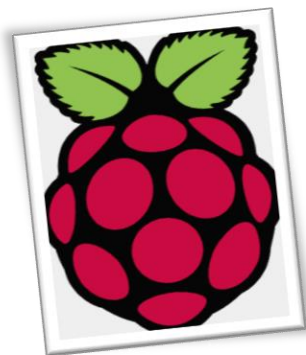
Inizia in questo numero una miniserie di 3 dedicata alla programmazione dell'Atari 2600. Scopriamo insieme a Giorgio Balestrieri l'hardware di questa macchina prima di poterci addentrare nella sua programmazione.

Articolo a pagina 11

Mame, Raspberry Pi e Jamma

Tramite una configurazione tutta manuale, niente di preconfezionato quindi, **Dante Profeta** ci mostra come installare e configurare un cabinet da sala giochi (Mame) utilizzando il Raspberry PI e la scheda Jamma.

Articolo a pagina 2

Mame, Raspberry Pi e Jamma

di Dante Profeta

Mame, Raspberry Pi e i Cabinati Arcade con Monitor CRT da 15 KHz.

...perché il retro-arcade è una cosa seria

Un paio di settimane prima della scrittura di questo articolo ho ricevuto una telefonata da un caro amico che non sentivo da tempo, felicissimo di avere acquistato il suo primo cabinet da sala giochi di tipo prontoscheda. Memore dei miei successi di oltre dieci anni addietro col mio prontoscheda, mi chiedeva supporto per installare e configurare un PC con Mame.

Oltre un decennio fa in effetti l'unica soluzione per avere Mame su un cabinet arcade era quella di imbottirlo con un PC di fascia medio alta, e di affidarsi a soluzioni come J-Pac e ArcadeVga o affini per l'interfacciamento col connettore Jamma. Di acqua ne è passata sotto i ponti da allora, e altre soluzioni come Pandora's sono emerse grazie all'aumentare della potenza di calcolo ed ad un'integrazione sempre più vicina al limite fisico consentito.

Tuttavia a me piace percorrere le strade meno battute e così ho colto l'occasione per sperimentare.

Conoscevo già da tempo l'adattatore VGA per Raspberry su connettore GPIO noto col nome di Gert's VGA Adapter. Questo adattatore ha il vantaggio di utilizzare le caratteristiche di Raspberry di pilotare un certo numero di pin del connettore GPIO per presentare in output i segnali RGB e modularne la frequenza. È noto infatti che i monitor CRT arcade accettano in ingresso solo segnali TTL a 15KHz, e l'uscita HDMI nativa mal si presta per questo scopo, se non al prezzo di pagare il downscaling, per me orrendo e non proponibile come soluzione tout-court.

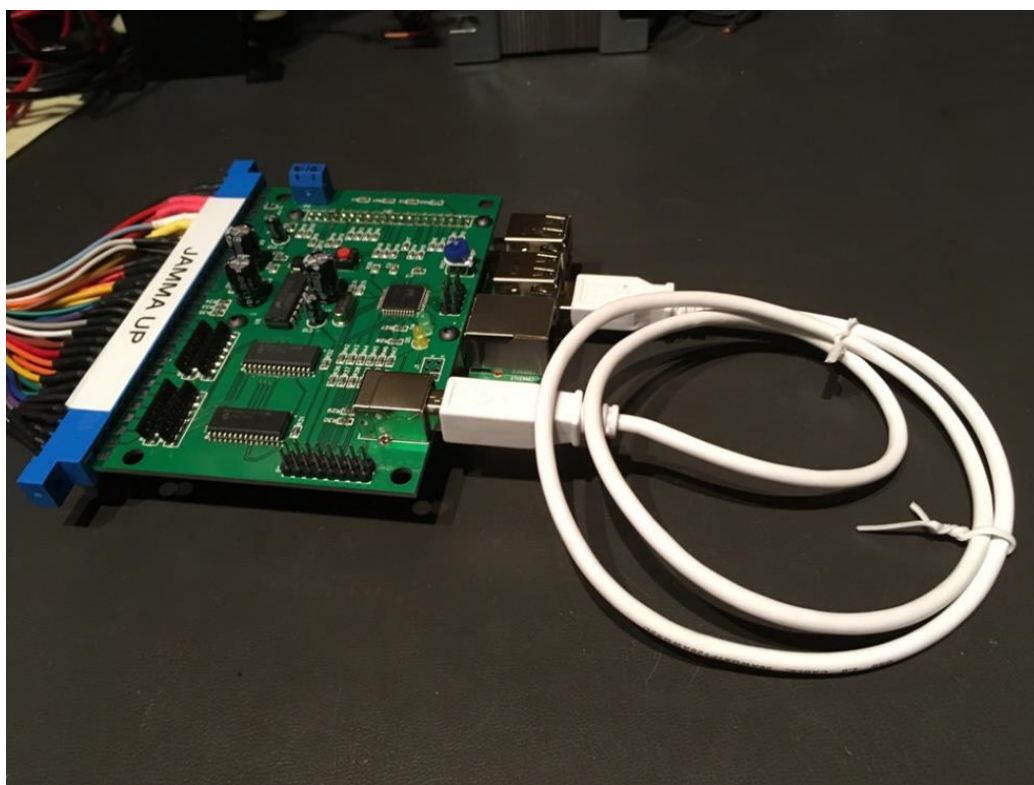
Dal 2007 bazzico, anche se con frequenza sporadica, il forum ArcadeItalia, e il caso volle che qualche settimana prima della telefonata del mio amico mi imbattessi proprio nel thread dell'adattatore Raspberry Jamma dell'ottimo Vincenzo Bini, basato proprio sul Gert VGA e con uscita video a 15KHz nativa.

L'idea di mettere assieme Raspberry Pi 3 e l'adattatore Jamma di Bini mi è subito piaciuta e aspettavo l'occasione giusta per sperimentarla e verificarne i pregi e i limiti.

Ordinati dunque i pezzi necessari ho fatto i primi esperimenti con la distribuzione RetroPie, che però in contrasto con le buone intenzioni di essere semplice da installare, si è verificata un inferno da customizzare con le configurazioni che avevo in mente, soprattutto per via di uno strano bug che non mi permetteva di mappare i pulsanti del pannello di gioco visti attraverso il controller emulato dalla scheda Jamma del bravo Vincenzo Bini.

Non mi sono mai piaciuti gli ambienti chiusi, belli e pronti, dove tutto è preconfezionato, perché si comportano come treni posti su rotaie, diventando ostici se non addirittura refrattari a qualunque tipo di deviazione dal percorso prestabilito.

Volendo dunque restare un semplice utente consumatore, ma con un bagaglio di esperienze trentennale come sistemista Unix, con delle esigenze del tutto personali, e non avendo intenzione di fare il salto della staccionata diventando un contributore attivo del progetto RetroPie, ho rapidamente deciso di abbandonarlo e di ripartire dalla distribuzione standard Raspbian di base e installare e configurare ciò che è strettamente necessario per far funzionare il solo Mame sul cabinet.

Questo articolo è dunque il resoconto fedele di come configurare un Raspberry Pi 3 con l'adattatore Jamma di Bini per giocare al meglio con Mame sul cabinet da

bar.

Cosa ci occorre

Oltre al Cabinet Arcade con Monitor CRT da 15 KHz o un provaschede, anche noto col nome di Supergun, da collegare a un televisore rigorosamente a tubo catodico, ci occorre un Raspberry Pi 3 modello B, una microSD e una scheda Raspberry Jamma di Vincenzo Bini.

Per le configurazioni iniziali una-tantum ci occorre una tastiera USB, un monitor HDMI, una connessione di rete, e un computer o laptop dotato di un dispositivo per la lettura/scrittura di SD o di microSD.

Raspbian

Per prima cosa occorre scaricare Raspbian sul proprio computer e copiarla su microSD come descritto sul sito di Raspberry Pi. La distribuzione che a noi serve è quella Stretch Lite, ovvero senza fronzoli e soprattutto senza server X11, ovvero ancora senza interfaccia grafica ma con la sola shell a linea di comando.

L'installazione di Raspbian Stretch Lite su microSD dipende dal S.O. e dal computer che utilizzate. Sul sito di Raspberry Pi nella sezione Download è ben documentata.

Il mio suggerimento è quello di utilizzare una microSD da 32 GB tra quelle supportate per contenere un set completo di ROM di Mame, esclusi i CHD si intende. Dimensioni inferiori risulterebbero insufficienti per contenerle tutte, mentre per includere anche i CHD non basterebbe una microSD da 64 GB.

La prima cosa da fare è quella di collegare Raspberry a un monitor HDMI, ad una tastiera USB, e alla LAN domestica con un cavo di rete o una connessione Wi-Fi. Il mouse non ci serve.

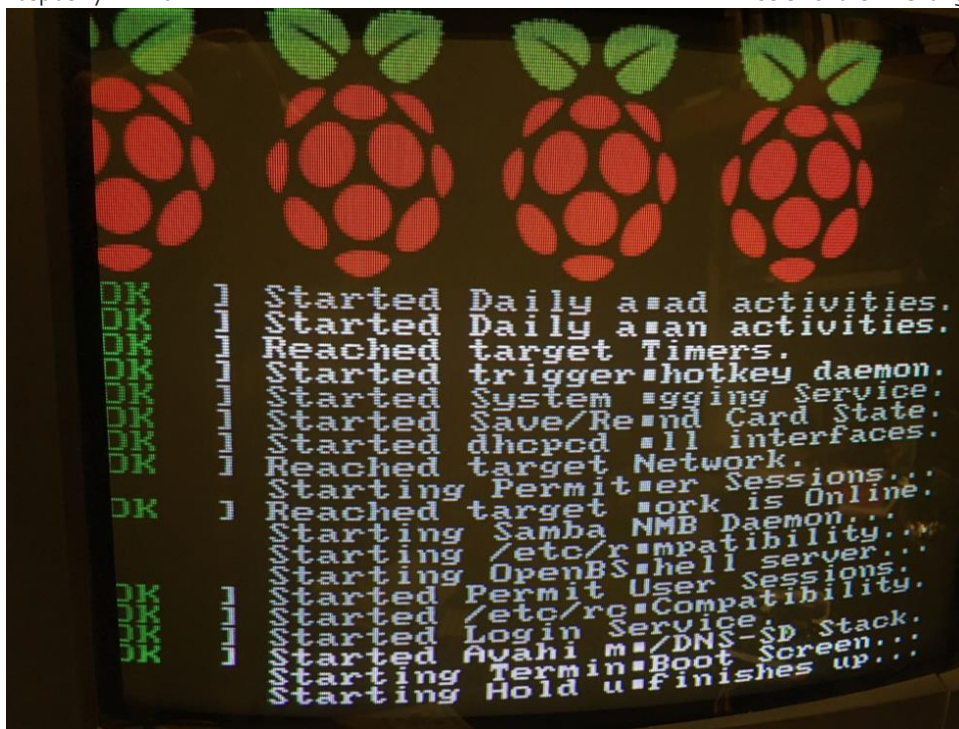
Inserita la microSD con Raspbian nel Raspberry Pi 3, al primo boot il sistema provvederà ad espandere il filesystem ext3, utilizzando tutto lo spazio disponibile sulla microSD.

Raspbian si riavvierà in automatico e ci proporrà il classico confortante prompt col cursore lampeggiante tipico delle shell Unix.

Raspberry, e Raspbian, per natalità, hanno un cuore che batte per la bandiera anglosassone, e lo dimostrano immediatamente col layout di tastiera la localizzazione tutta in English UK.

Non so voi, ma io che sono nato e cresciuto con il Commodore 64 prima, e con Amiga poi, ho sempre apprezzato il layout delle tastiere americane, e non avendo mai dovuto fare il dattilografo, ma il sistemista/programmatore per professione, ho sempre ricercato e utilizzato solo tastiere americane English US.

Il vantaggio delle tastiere English US è quello di avere mappati direttamente i tasti dei caratteri speciali ~, {, }, [,], necessari per la navigazione su shell Unix, per la programmazione script sh, e per i linguaggi C-style.

La ~ (tilde) ad esempio, sotto Unix significa Home, e ci consente di fare sempre riferimento alla nostra home directory. So che è possibile ottenere questo e gli altri caratteri su PC con delle combinazioni di tasti, ma non avendo mai utilizzato tastiere italiane non mi sono mai posto il problema di imparare le sequenze.

Il nostro primo passo nelle configurazioni di base sarà dunque quello di localizzare Raspbian per il nostro ambiente locale, cominciando dal TimeZone, passando dal linguaggio di sistema settando opportunamente una variabile d'ambiente necessaria per l'installazione e l'utilizzo dei programmi, e finendo al layout di tastiera, per me English US, ma voi potete anche metterlo in Italiano standard se preferite e se vi è più congeniale, consapevoli delle limitazioni su descritte. Ecco i passaggi necessari.

```
> sudo raspi-config
```

Entrate nel submenu "Localisation Options". Cominciamo con impostare l'ora corretta:

Selezionate "Change Timezone"; quindi "Europe" e infine navigate fino a "Rome". Selezionate "Ok" e tornate al menu` di Localisation.

Selezionate "Change Locale", deselezionate la voce "en_GB.UTF-8 UTF-8" e scegliete invece "en_US.UTF-8 UTF-8". Questo passaggio e` importante che sia così perché verrà riflesso nella variabile d'ambiente LANG che serve per diverse ragioni.

Infine è il momento tanto atteso della tastiere:

Sempre da "Localisation Options", selezionare "Change Keyboard Layout", quindi "Generic 105-key (Intl) PC"; ci apparirà un elenco di possibili tastiere inglesi che a noi non interessano, pertanto scendete giù fino a selezionare "Other".

Per chi come me utilizza le tastiere americane:

scegliere "English (US)".

Per chi invece desidera il layout di tastiera italiano:

scegliere "Italian".

A questo punto, visto che siamo nel menù di configurazione di Raspbian è opportuno abilitare l'autologin. Lo utilizzeremo per far partire automaticamente Mame con privilegi da utente invece che con privilegi root se invocato ed eseguito attraverso gli script rc di boot, ovvero gli script che vengono eseguiti all'avvio contenenti l'elenco dei programmi da avviare in background a corollario del S.O.

Dal menu principale selezionare "Boot Options", quindi "Desktop / CLI" e infine "Console Autologin".

Dopo un sano e necessario reboot, col comando "sudo reboot", per riflettere le modifiche apportate alle variabili d'ambiente e verificare il login automatico senza bisogno di inserire username e password, il passo

successivo è quello di aggiornare Raspbian all'ultima versione con gli ultimi bug disponibili. Ma prima è necessario connetterci alla rete domestica.

Se si dispone di un cavo di rete l'operazione è immediata per via del DHCP che ci procurerà al volo un indirizzo IP, altrimenti per il Wi-Fi occorrerà effettuare le opportune configurazioni come descritto puntualmente sul sito di Raspberry Pi e riportato come link in calce tra i riferimenti di questo articolo. Io personalmente mi affido sempre al cavo di rete se è possibile.

Stabilita la connessione alla rete (è sempre bene verificarla con un ping ad esempio al dns primario di google: "ping 8.8.8.8") è bene procedere con l'aggiornamento del S.O.

```
> sudo apt-get update && sudo apt-get upgrade
```

```
> sudo reboot
```

La configurazione di base procede con l'istruire Raspbian a non attendere all'avvio che sia disponibile una connessione di rete per continuare il boot, altrimenti, a cavo di rete disconnesso dovremo aspettare un certo quantitativo di secondi prima che la ricerca di una connessione di rete vada in timeout, rallentando anche di un minuto il tempo di avvio di Raspbian.

```
> sudo raspi-config
```

Selezionare la voce "Wait for Network at Boot" e scegliere "No" nella schermata successiva.

Configurazioni per la gestione remota

Al fine di consentire l'amministrazione di Raspberry da remoto dovremo abilitare SSH e assegnare un indirizzo IP statico per praticità. Ma queste operazioni sono opzionali visto che richiedono la conoscenza di un editor di testo a linea di comando come vi, e potrebbero risultare ostiche a qualcuno di voi.

È comodo, e come vedremo più avanti può risultare indispensabile, abilitare SSH per l'amministrazione da remoto, cosicché non avremo necessità alcuna di infilare una scomoda tastiera dentro il cabinet e dover ricalibrare ogni volta l'area visibile sul monitor CRT per contenere l'intera shell. Con SSH faremo semplicemente uso della connessione alla rete locale domestica, oppure una connessione punto-punto col nostro laptop/pc preferito, nel caso in cui debba essere necessario rimettere mano alle configurazioni o sistemare qualche problema.

```
> sudo raspi-config
```

Selezionare "Interfacing Options", quindi "SSH" e rispondere "Yes" alla schermata successiva.

Sarebbe anche opportuno istruire Raspbian a prendere un indirizzo IP statico, che definiamo noi, nel caso non riuscisse ad ottenerlo via DHCP. In questo modo potremo sempre raggiungerlo via rete e soprattutto risulterà utile nella connessione punto-punto col laptop o col pc.

Per far ciò è necessario l'utilizzo di un editor a linea di comando. Io da sempre uso vi, che richiede però una trattazione a sé, pertanto vi suggerisco di provare a usare "nano" o qualche altro editor a voi più familiare, al fine di aggiungere le seguenti linee nel file di configurazione /etc/dhcpcd.conf, sostituendo gli indirizzi IP corretti alle scritte in grassetto:

```
# It is possible to fall back to a static IP if DHCP fails:

# define static profile

profile static_eth0

static ip_address=indirizzo_ip_preferito_del_raspberry/24

static routers=gateway_di_rete

static domain_name_servers=8.8.8.8


# fallback to static profile on eth0

interface eth0

fallback static_eth0
```

Poiché faremo in modo che alla partenza di Raspbian, all'autologin dell'utente di default pi, parta automaticamente mame, avremo bisogno di un account di servizio con privilegi sudo per fare login da remoto in caso di necessita.

Creiamo prima l'utente "service"

```
> sudo adduser service
```

Imporre una password per l'utente "service" e confermarla; quindi fornire le informazioni opzionali sull'utente.

Attribuiamogli la possibilità di eseguire comandi come root (sudo).

```
> usermod -aG sudo service
```

Testiamo che riusciamo a fare login come "service"

```
> su - service
```

Immettiamo la password che abbiamo scelto prima; una volta dentro come "service" torniamo al nostro utente "pi" di default:

```
> exit
```

Il Pendrive USB come strumento per l'import dei file di configurazione:

Per chi avesse difficoltà nell'utilizzo di vi, suggerisco di editare o creare i file di configurazione con notepad (blocco note) o simile su Windows, e con TextEdit o simile su OS X avendo l'accortezza, con quest'ultimo strumento, di salvare i file di testo in formato UTF-8.

In ogni caso, per leggere e scrivere sui pendrive USB può essere necessario installare i driver exfat. Ricordate? Abbiamo scelto la versione Lite di Raspbian per avere giusto l'essenziale, e alcune cose come questa mancano e dobbiamo installarle a mano:

```
> sudo apt-get install exfat-fuse
```

A questo punto per montare il pendrive dobbiamo utilizzare il comando:

```
> sudo mount /dev/sda1 /mnt
```

Per rimuoverlo occorrerà invocare il comando:

```
> sudo umount /mnt
```

Mame, cioè AdvanceMame

La versione di Mame che consiglio di installare è l'italianissima AdvanceMame per Raspberry Pi.

AdvanceMame non ha certo bisogno di presentazioni, ma per chi non la conoscesse dirò brevemente che si tratta di una versione ottimizzata di Mame pensata per girare al meglio con i monitor CRT.

Al momento della scrittura di questo articolo l'ultima versione disponibile è la 3.6, e sebbene sul sito viene riportato che è basata su Mame 0.106, il romset completo richiede almeno la versione 0.146.

Per installarla occorre scaricare l'archivio dal sito ufficiale e utilizzare gli strumenti di installazione Debian. Il link qui riportato si riferisce alla release 3.6,e potrebbe certamente cambiare in futuro.

```
> cd ~
```

```
> wget https://github.com/amadvance/advancemame/releases/download/v3.6/advancemame_3.6-1_armhf.deb
```

```
> sudo dpkg -i advancemame_3.6-1_armhf.deb
```

Gli eseguibili advmame e advmenu verranno posti nella directory /usr/local/bin, mentre tutti gli altri file e directory, eccetto quelli di configurazione, verranno posti in /usr/local/share/advance. I file di configurazione troveranno posto invece nella

directory nascosta ".advance" della nostra home directory.

Come dicevo, nella directory /usr/local/bin troveremo sia advmame che advmenu. Advmenu è un frontend senza fronzoli ma efficace per la selezione dei giochi da eseguire con advmame, che trae vantaggio dagli snapshot e dai filmati mp4 per mostrare al giocatore un breve filmato del gioco che si sta selezionando. La versione 3.6 di advmenu mostra tuttavia un bug audio, almeno nella mia versione di Raspbian, che genera un fruscio di fondo simile alle arancine in frittura quando il filmato torna in loop. Roba di poco conto, e comunque esistono altri frontend grafici più accattivanti come Cabrio, oppure Wah!Cade, clone in python del ben noto MameWAH.

Nell'archivio sono presenti alcune rom i cui diritti sono stati modificati dai rispettivi autori per renderle fruibili al vasto pubblico.

A questo punto possiamo effettuare i primi test e goderci i risultati. Per prima cosa eseguiamo advmame:

```
> advmame gridlee
```

e poi testiamo anche advmenu:

```
> advmenu
```

Le ROM, i Sample, e gli Snapshots -SAMBA

Nella directory advmame troviamo le solite sottodirectory tipiche di mame. Quelle che a noi interessa popolare sono: ./roms ./samples e ./snap

Come fare? Ci sono diversi metodi, ma quello che ritengo più pratico è di installare un file server SMB, ovvero SAMBA, e montare la directory advmame con accesso pubblico in lettura e scrittura per vederla su Windows o OS X come un folder remoto.

Per fare ciò dobbiamo prima installare SAMBA:

```
> sudo apt-get install samba samba-common-bin
```

La configurazione richiede l'edit del file di configurazione /etc/samba/smb.conf, per aggiungere le seguenti righe:

```
[share]
Comment = AdvanceMame shared folder
Path = /usr/local/share/advance
Browseable = yes
Writeable = Yes
only guest = no
create mask = 0777
directory mask = 0777
Public = yes
Guest ok = yes
```

A questo punto occorrerà creare una password samba per l'utente pi:

```
> sudo smbpasswd -a pi
```

Infine avviamo samba e lo rendiamo sempre disponibile:

```
> sudo /etc/init.d/samba restart
```

Per accedere alla directory condivisa da Windows o da OS X è necessario conoscere l'indirizzo IP che ci è stato assegnato dal DHCP oppure verificare quello che abbiamo assegnato noi staticamente col comando:

```
> ifconfig -a
```

Su Windows

Premere contemporaneamente i tasti Windows e R; apparirà la finestra Esegui in cui dovremo scrivere:

`\\indirizzo_ip_del_raspberry` e premere invio.

Su OS X

Dal menu "Go" del Finder selezionate "Connect to Server" e immettete la stringa:

```
smb://indirizzo_ip_del_raspberry
```

Comparirà una finestra di autenticazione dove dovremo inserire come username "pi" e come password quelle che abbiamo scelto quando abbiamo invocato il comando smbpasswd su Raspbian.

A questo punto possiamo copiare le ROM di Mame nella cartella roms; i samples nella cartella samples; gli snapshots nella cartella snap.

E con questo abbiamo completato il setup di AdvaceMame, a meno delle configurazioni dei pulsanti e dei joystick, che vedremo tra un po'.

La scheda Jamma

È finalmente giunto il momento di dedicarci alla scheda add-on Jamma che nel frattempo avremo montato sul connettore GPIO, a Raspberry spento e scollegato da qualsiasi cavo. Attenzione alla polarità e a non sbagliare a montarla per non correre il rischio di bruciare qualcosa.

Vi consiglio caldamente di montare la scheda sui distanziatori forniti nel kit per evitare che possa adagiarsi sul metallo dei connettori del Raspberry e creare un cortocircuito. I perni dei distanziatori che ho ricevuto io non entravano nei quattro fori del Raspberry, pertanto ho dovuto allargare leggermente i fori col trapano e fissare i perni sotto la pcb con dei dadi che tuttavia non sono forniti nel kit.

Per completare il montaggio colleghiamo il cavo USB in dotazione, che servirà per riportare a Raspberry i segnali Jamma dei joystick e dei pulsanti, facendo apparire la scheda Jamma come se fosse un controller da gioco.

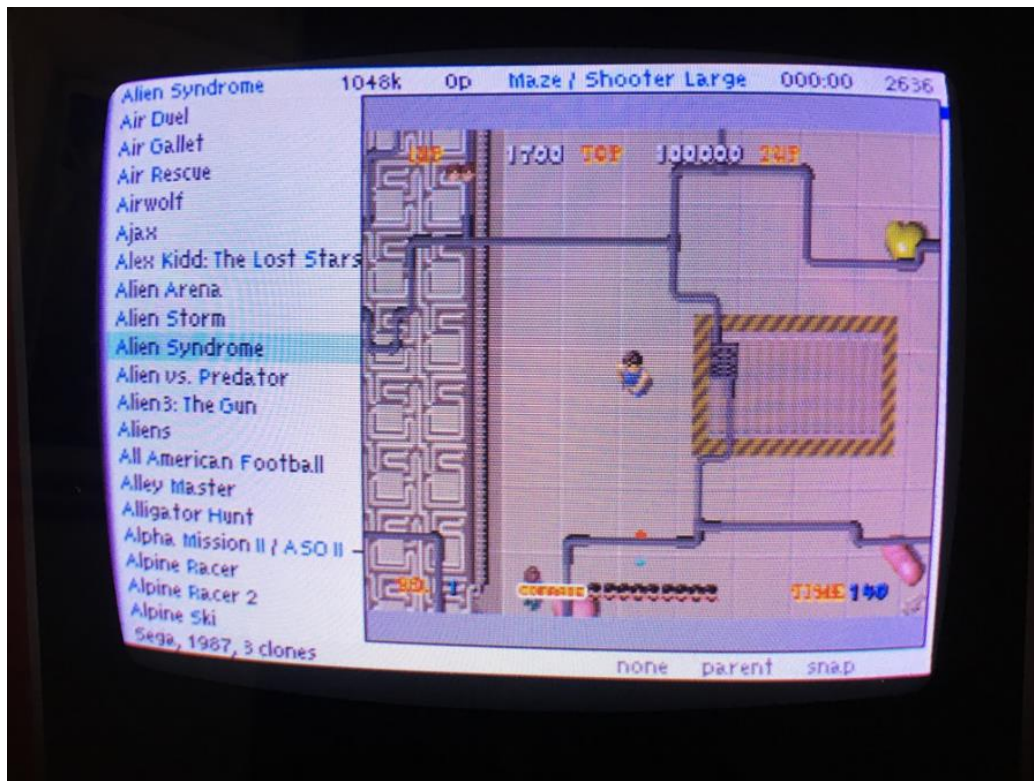
Jamma e i 15 KHz

Fino ad adesso abbiamo utilizzato l'uscita video HDMI e non ci siamo preoccupati di come gestire l'output video per la scheda Jamma.

Raspbian utilizza una struttura chiamata overlays per gestire le caratteristiche peculiari di Raspberry Pi. Delle sorte di driver, per semplificare. Gli overlays trovano posto nella directory /boot/overlays.

Assieme agli overlays esiste anche un file di configurazione denominato config.txt che risiede nella directory /boot e contiene i parametri di configurazione dell'hardware. I parametri contenuti nel file config.txt sono assimilabili ai parametri memorizzati nella CMOS del BIOS dei PC. Ad ogni modifica del file config.txt sarà necessario fare un reboot.

Per abilitare l'uscita RGB a 15KHz dobbiamo innanzitutto scaricare e copiare i files vga565-3.dtbo e vga666-6.dtbo nella directory overlays, e aggiungere le seguenti righe di configurazione nel file config.txt:

```
dtoverlay=pwm-2chan,pin=18,func=2,pin2=19,func2=2
dtoverlay=vga666-6
enable_dpi_lcd=1
```

```
display_default_lcd=1
dpi_output_format=6
dpi_group=2
dpi_mode=87
# Abilitare la riga di sotto per i Monitor con Sincronismo Positivo
#hdmi_timings=320 1 16 30 34 240 1 2 3 22 0 0 0 60 0 6400000 1
# Abilitare la riga di sotto per i Monitor con Sincronismo Negativo
#hdmi_timings=320 0 16 30 34 240 0 2 3 22 0 0 0 60 0 6400000 1
```

Il parametro hdmi_timing impone la risoluzione dello schermo e il segnale di sincronismo per il monitor.

Se il nostro monitor si aspetta un segnale di sincronismo positivo allora dovremo abilitare la riga sotto il commento Sincronismo Positivo, altrimenti, se il nostro monitor si aspetta un segnale di sincronismo negativo dovremo abilitare la riga sotto il commento Sincronismo Negativo.

L'abilitazione di una delle due righe è critica. In primo luogo perché appena abilitiamo una delle due escludiamo automaticamente l'uscita video sul connettore HDMI, e quindi non vedremo più nulla sul monitor HDMI che abbiamo ancora collegato.

In secondo luogo per via della scelta tra sync positivo o negativo, perché se abilitiamo il sincronismo errato vedremo sul monitor del cabinet un'immagine fuori sincrono e illeggibile, e non saremo più in grado di editare il file di configurazione.

Detto questo, le elettroniche dei monitor con cui ho sempre avuto a che fare si aspettano un sincronismo positivo, quindi, a meno che non abbiate elettroniche esotiche come quelle dei cabinet Nintendo, è molto probabile che anche il monitor del vostro cabinet abbia bisogno di un segnale di sincronismo positivo.

Tuttavia se non si conosce il tipo di segnale di sincronismo accettato occorrerà procedere per tentativi, che comunque non danneggeranno né il monitor né Raspberry né la scheda Jamma.

Se abbiamo sbagliato a scegliere il tipo di segnale di sincronismo, e se abbiamo abilitato precedentemente SSH basterà accedere a Raspbian da remoto con un client SSH tipo Putty per Windows, o col comando ssh da terminale su OS X, per andare a modificare il file config.txt commentando la riga di sincronismo errata e abilitando l'altra, seguito da un necessario reboot del sistema.

Se invece non abbiamo previsto un accesso da remoto a Raspberry allora siamo costretti ad estrarre la scheda sd, connetterla a un pc, accedere alla partizione /boot che per nostra fortuna è in fat32, ed editare da qui il file config.txt.

Prendete dunque le vostre brave precauzioni prima di procedere.

A questo punto, se abbiamo scelto correttamente il segnale di sincronismo del monitor ci resta un'ultima operazione per la messa a punto del controller integrato nella scheda Jamma: occorre editare il file cmdline.txt e aggiungere "usbhid.quirks=0x2341:0x8037:0x040"

```
dwc_otg.lpm_enable=0 usbhid.quirks=0x2341:0x8037:0x040 console=ttyAMA0,115200 console=tty1 root=/dev/mmcblkop2 rootfstype=ext4 elevator=deadline rootwait
```

Abbiamo praticamente finito, almeno col grosso dell'installazione..

Spegniamo Raspberry invocando la sequenza di shutdown:

```
> sudo shutdown -h now
```

Aspettiamo qualche istante che Raspbian completi la sequenza di shutdown, stacchiamo il cavo HDMI e colleghiamo il connettore Jamma al cabinet prontoscheda. Manteniamo collegate la tastiera e il cavo di rete perché ci serviranno ancora.

Diamo corrente al cabinet. Dovremmo riuscire a vedere correttamente l'intera sequenza di boot e infine la shell Raspbian col cursore bianco lampeggiante.

Per curiosità e appagamento per gli sforzi fin qui fatti eseguiamo baldanzosi advmame o advmenu e facciamo un test veloce, per scoprire che dobbiamo ancora mappare correttamente i pulsanti del pannello e le gettoniere.

Configurazione del pannello di controllo

Avviamo advmame con una rom a vostra scelta. Nell'esempio seguente eseguirò Mame con Metal Slug, di cui possiedo la cartuccia originale e la base NeoGeo, anche per fare un semplice confronto.

```
> advmame mslug
```

Tenendo la tastiera a portata di mano premiamo il tasto Tab ed entriamo nel menù di configurazione dei dispositivi di Input. Dobbiamo mappare i joystick, i pulsanti e le gettoniere.

Fatta questa operazione torniamo al menu iniziale con Esc ed entriamo nel menu "input general/User Interface" e alla voce "UI Cancel" mappiamo una sequenza di tasti per uscire da advmame e tornare su advmenu per scegliere un altro gioco a cui giocare.

Sul cabinet a tre pulsanti del mio amico io ho scelto la pressione contemporanea dei tre pulsanti del primo giocatore.

Un problema che ho avuto è stato di non riuscire a far salvare le configurazioni se immesse dal menu General Input. Se anche voi avete lo stesso problema vi suggerisco il seguente workaround:

Anziché configurare sotto General Input scegliete la configurare specifica per il gioco in esecuzione.

Finite le configurazioni tornate al prompt dei comandi e andate nella directory nascosta ".advance" nella vostra home directory:

```
> cd ~/.advance
```

editate il file di configurazione di AvanceMame:

```
> vi advmame.rc
```

Andate fino in fondo al file muovendovi con Ctrl+d; vedrete le vostre configurazioni cominciare con "mslug/input_map". Rimuovere il prefisso "mslug/" dalle righe facendo in modo che inizino tutte per "input_map" e avrete esportato la vostra configurazione da mslug a globale. Salvate e uscite da vi (:x!)

Boot e autopartenza di AdvanceMame

Ci sono almeno due modi per avviare advmenu al boot. Un metodo è quello di eseguirlo nei file rc di sistema. Questa soluzione la trovo davvero brutta soprattutto perché verrebbe eseguito con privilegi di root.

Il metodo che ho scelto e che preferisco è quello di lanciare advmenu con privilegi utente. Se abbiamo abilitato l'autologin in console per l'utente pi, ciò che dobbiamo fare è aggiungere le seguenti righe alla fine del file nascosto ~/.bashrc

```
# start AdvanceMenu
exec /usr/local/bin/advmenu
```

Shutdown?

Ed eccoci infine a una problematica interessante, ovvero come spegnere Raspberry e il cabinet.

Non so se avete posseduto una Amiga con Hard Disk, ma se lo avete avuto vi ricordate di come facevamo lo shutdown? Lo spegnevamo dal pulsante di spegnimento togliendo la corrente. Non c'era nessuna procedura di shutdown. Ma allora perché adesso invece tutti i dispositivi con un S.O., Raspberry incluso, richiedono una procedura di shutdown? È possibile avere un sistema che si possa spegnere semplicemente togliendo corrente come facevamo tanti anni fa? Cosa è cambiato da allora ad ora?

Principalmente ciò che è cambiato nei S.O. moderni è che è stato aggiunto il gestore di memoria virtuale, che serve per utilizzare la memoria di massa come supporto per lo swapping delle pagine dei processi, al fine di poter far girare nella RAM più programmi possibile e anche dati più grandi come ad esempio molteplici immagini raw.

Raspberry Pi 3 dispone di 1GB di RAM, pertanto per far girare X11, ovvero l'interfaccia a finestre, e i programmi come i word processor o i browser web, 1GB potrebbe non bastare di per se, e per tale ragione ci si appoggia sulla memoria virtuale.

Per i giochi che andremo ad emulare con Mame ricadiamo al di sotto delle centinaia di Mega Bytes. Ad esempio i giochi Neo-Geo potenzialmente possono raggiungere al più i 330 Mbit ovvero 41.25 MByte.

Possiamo disabilitare completamente il gestore di memoria virtuale, liberandoci così dalla schiavitù della procedura di shutdown:

```
> sudo dphys-swapfile swapoff
> sudo dphys-swapfile uninstall
> sudo update-rc.d dphys-swapfile remove
```

Poiché Linux tiene traccia di tutti gli eventi di sistema, come gli errori, in appositi file di log contenuti nella directory /var/log, dobbiamo fare in modo che la scrittura su questi file non possa causare corruzioni del filesystem nel caso in cui Raspberry venga spento togliendo corrente. Per far questo è opportuno montare la directory /var/log come filesystem temporaneo, ovvero in RAM. Ovviamente i file di log non sopravviveranno al reboot, ma poco ci importa per i nostri scopi. Dobbiamo dunque aggiungere questa riga nel file /etc/fstab:

```
tmpfs /var/log tmpfs size=1M,noatime 0 0
```

Con questi due accorgimenti possiamo spegnere Raspberry togliendo la corrente in qualsiasi momento senza preoccupazione di corrompere il filesystem sulla microSD, e vivere un esperienza simulata ogni volta come se fosse davvero una macchina dedicata da bar.

Pixel Perfect:

Ed eccoci alla domanda clou. Perché utilizzare Raspberry Pi 3 al posto di un box Pandora's, o di un PC con ArcadeVGA o affine? Perché l'uscita GPIO di Raspberry può essere riconfigurata al volo per simulare al meglio persino le risoluzioni native delle schede originali da bar, evitando gli orribili scaling necessari se si mantiene sempre la stessa risoluzione a prescindere da quella del gioco. Fantastico!

Come fare? E` un'operazione abbastanza semplice grazie all'utility advv di AdvanceMame, ma per i dettagli vi rimando alla bellissima guida a riguardo descritta sul forum ArcadeItalia dal bravo Yami, di cui trovate il riferimento in calce a questo articolo.

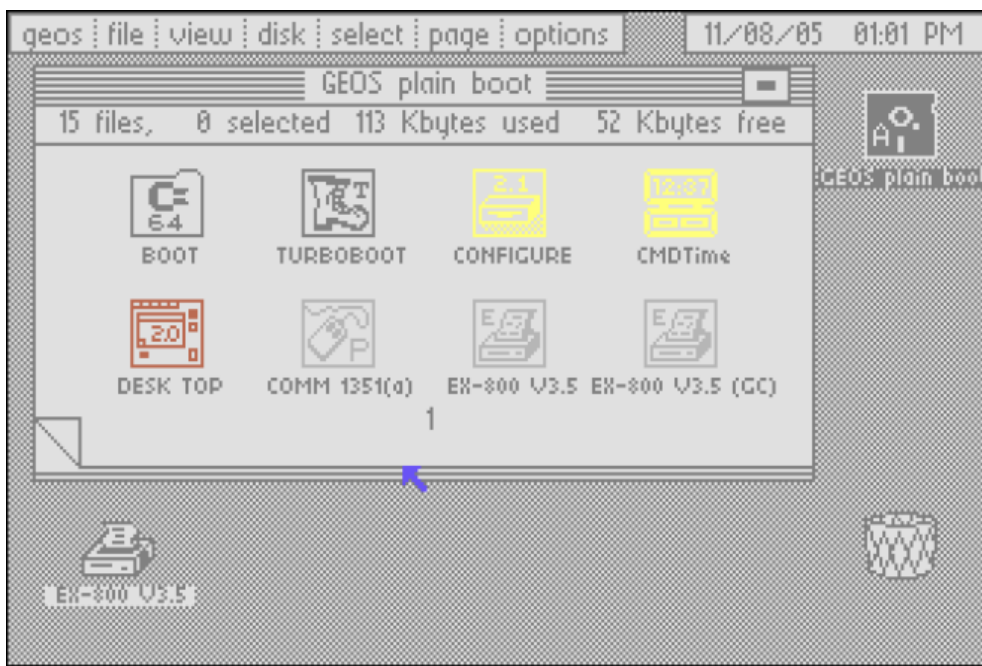
Insomma, Raspberry Pi 3 ha davvero una marcia in piu`, che assieme alla scheda Jamma di questo articolo, lo rende unico come dispositivo di emulazione per i giochi dei cabinati da sala giochi.

Riferimenti

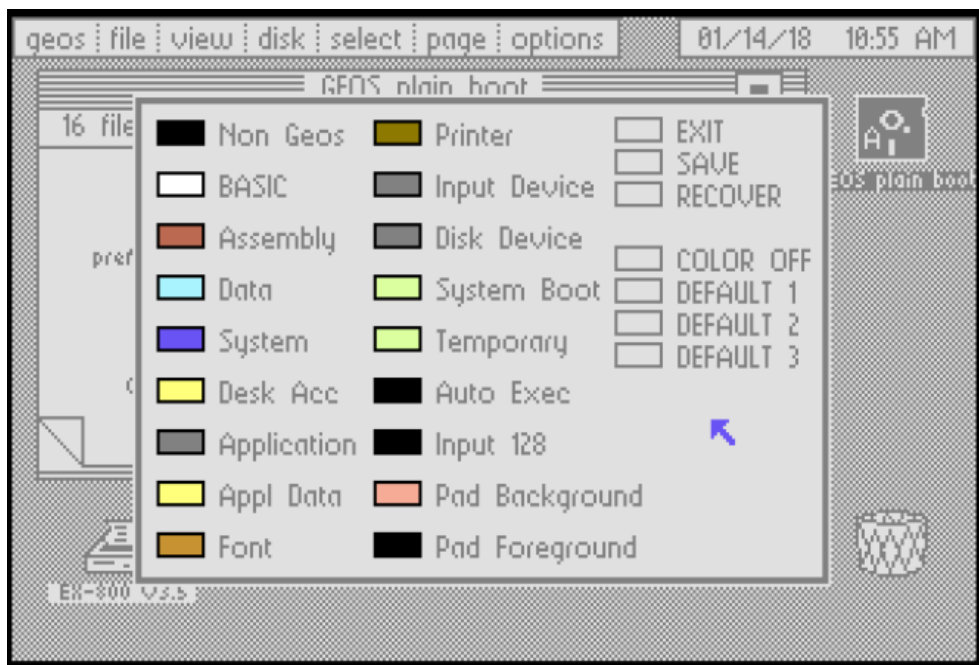
Raspberry Pi:
https://www.raspberrypi.org
Gert's VGA Adapter:
https://www.raspberrypi.org/blog/gert-vga-adapter/
ArcadeItalia:
http://www.arcadeitalia.net
Scheda Jamma:
http://www.jammapi.it
Raspbian Stretch Lite:
https://www.raspberrypi.org/downloads/raspbian/
Configurazione Wi-Fi:
https://www.raspberrypi.org/documentation/configuration/wireless/wireless-cli.md
AdvanceMame:
http://www.advancemame.it
Overlays:
https://www.jammapi.it/wb/media/overlay.zip
Guida Pixel-Perfect:
http://arcadeitalia.net/viewtopic.php?f=79&t=29557

Il GEOS del Commodore 64…

di Francesco Fiorentini

Era il 1987, avevo il Commodore 64C da circa un annetto e sino ad allora lo avevo utilizzato prevalentemente per giocare, quando, spinto dalla curiosità di comprendere come questa macchina 'magica' funzionasse, decisi di cominciare a documentarmi ed a leggere le riviste del settore dell'epoca. In quel periodo, a parte i giochi che offrivano agli occhi di un ragazzino milioni di colori ed un sonoro emozionante, la quasi totalità delle applicazioni che mi era passata per le mani utilizzava l'interfaccia a caratteri tipica del C64 con pochissimi vezzi grafici. Non ricordo dove lo vidi, forse in una delle riviste sopra menzionate oppure nella vetrina di uno dei negozi di informatica che frequentavo all'epoca, fatto sta che questo programma attrasse la mia attenzione come fa la luce con le falene! Il software in questione era il GEOS (acronimo di Graphic Environment Operating System) e lì per lì mi sembro' la cosa più evoluta che avessi mai visto farere al mio C64! Lo volevo, dovevo averlo! Dovevo mettere le mani su quel programma che mi avrebbe catapultato nel futuro. Ma come? IL GEOS veniva distribuito solo su dischetti da 5.25 ed io non avevo il lettore floppy… Decisi quindi di cominciare a risparmiare un po' di soldi ed alla fine, grazie anche all'aiuto di babbo e mamma riuscii finalmente a comprare il drive 1571 (quello che si confaceva al mio C64C) ed allo stesso tempo ottenere l'agognato GEOS. Inizialmente usai il GEOS con il joystick, ma da lì a poco decisi di mettere di nuovo mano al portafoglio ed acquistare anche il mouse che era lo strumento principe per utilizzare il suddetto sistema operativo…

Scusate il lungo preambolo, ma sono dell'idea che accompagnare un articolo tecnico con un ricordo della nostra infanzia lo faccia diventare più umano e quindi più godibile.

Secondo me il GEOS è stato una vera e propria killer application per il Commodore 64, al quale per la prima volta forniva un'interfaccia grafica (**GUI** acronimo di Graphic User Interface) ed allo stesso tempo implementava funzionalità tipiche dei computer di fascia alta e mai viste prima sull'8 bit della Commodore. Stiamo parlando di un vero e proprio sistema operativo che oltre a ridefinire l'interfaccia nativa del C64, permetteva l'accesso ed il controllo di tutte le periferiche connesse tramite degli appositi driver. Come dicevo, complice anche la diffusione del Commodore 64, il GEOS è stato per un breve lasso di tempo il terzo più popolare sistema operativo dietro soltanto a DOS e Mac OS. Un risultato di tutto rispetto.

Come siano riusciti i maghi della Berkeley Softworks a far girare su una CPU a 8bit da 1Mhz e con 64Kb di RAM, di cui soltanto 38kb liberi, un SO complesso come il GEOS, si può spiegare solo con un'attenta ottimizzazione del codice. Il software inoltre era dotato di un complesso sistema di protezione dalla copia, reso necessario proprio per proteggere l'ottimo lavoro svolto dai programmatori.

Per i più curiosi c'è un interessante articolo pubblicato qui: http://www.pagetable.com/?p=865

L'articolo tratta proprio delle trappole anticopia inserite nel codice del GEOS ed in alcuni suoi applicativi. Vi suggerisco di leggerlo, è veramente affascinante. Per coloro che fossero interessati all'articolo ma trovassero difficoltà con l'idioma di Albione, potete inviarci una mail all'indirizzo della redazione ed in caso di un numero sufficiente di richieste, potremmo pensare di farne una traduzione.

Come dicevo prima, il GEOS è un Sistema Operativo vero e proprio che riconfigura completamente il Kernal del C64, implementando nuove primitive per gestire la grafica in alta risoluzione, l'I/O delle periferiche e l'interfacciamento con l'utente. Ma vediamo un po' quali sono le nuove caratteristiche implementate dal GEOS e spendiamo due parole su ognuna di esse:

- *Interfaccia gestita tramite mouse*
- *Drag and drop per le operazioni con i file*
- *Implementazione del WYSIWYG*
- *Sistema basato su driver*

Interfaccia gestita tramite mouse

Adesso diamo per scontato l'utilizzo del mouse e lo percepiamo come una periferica standard per i nostri computer, una sorta di estensione naturale della mano che permette di eseguire buona parte delle operazioni con cui controlliamo il PC, ma vi posso garantire che nel 1986/87, nonostante fosse stato inventato da circa 20 anni, il suo utilizzo era tutto tranne che scontato. Passare di punto in bianco dall'utilizzo della tastiera e del joystick al mouse non fu proprio una passaggiata. Vale inoltre la pena ricordare che i mouse dell'epoca non erano precisi e sofisticati come quelli attuali. Il Mouse del Commodore 64, il 1351, aveva una pallina pesantissima che scorreva su due rotelline che sembrava facessero a gara a raccogliere più sporcizia possibile ed i pulsanti, in quanto a praticità lasciavano alquanto a desiderare. Ironia a parte, l'innovazione dell'interfaccia punta e clicca del GEOS era ed è tuttora palpabile, finalmente il C64 si presentava all'utente con una veste tutta nuova e soprattutto con un ambiente più user friendly!

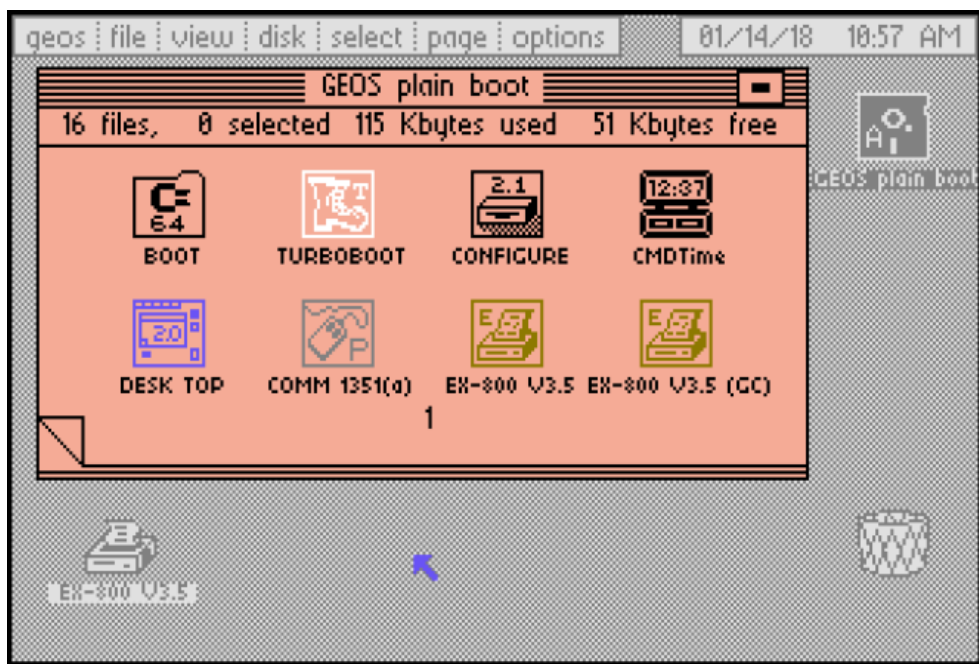
L'utilizzo del mouse risulta preciso e puntuale e l'interfaccia desktop, nonostante sia essenziale, e' ben organizzata e funzionale allo scopo. Decisamente un balzo in avanti di qualche anno rispetto alla media dei prodotti commerciali del C64. Pensate ad un utilizzo del biscottone in ambito commerciale, fino all'avvento del GEOS era appannaggio di pochi manager temerari, mentre con il GEOS era finalmente alla portata di tutti (almeno in linea teorica).

Una menzione particolare voglio farla per le operazioni eseguibili sui files. Scordatevi la tastiera, finalmente operazioni come la copia di un file possono essere eseguite tramite il comodo comando posizionato in alto allo schermo. Per duplicare un file bastera' selezionarlo e poi tramite il menu' **file**, scegliere il comando **duplicate**. Decisamente semplice non credete?

Drag and drop per le operazioni con i file

La funzionalità drang and drop, trascina e rilascia, è una diretta conseguenza del punto precedente. Provate a pensare alle istruzioni che avreste dovuto impartire da riga di comando in BASIC per stampare un semplice file sulla stampante; qualcosa di simile a:

OPEN 1,4:CMD 1:LIST:PRINT #1:CLOSE 1

Ecco immaginate invece di fare la stessa cosa trascinando il file sull'icona della stampante... Strabiliante vero? Su di un Commodore 64 nel 1986 decisamente si'!

Ma non e' tutto, provate per esempio a cliccare su un file ed a trascinarlo sul cestino. Esatto il file verra' cancellato, scomparendo dal disco dove era memorizzato. E per recuperarlo? Ovviamente bastera' cliccare sul cestino ed il fle verra' ripristinato.

Implementazione del WYSIWYG

WYSIWYG è l'acronimo di What You See Is What You Get - quello che vedi è quello che ottieni. In parole povere per la prima volta sul nostro home computer avevamo a disposizione la trasposizione 1-1 di quello che vedevamo su schermo. Per apprezzare a pieno questa caratteristica, non visibile tramite il DeskTop del GEOS e' necessario utilizzare il programma GeoWrite!

Con GeoWrite per la prima volta infatti si ha la netta sensazione di scrivere su un vero foglio A4 e la dimensione del foglio e' percepibile anche tramite la piccola mappa virtuale posizionata in alto al centro dello schermo dove, usando il mouse, e' possibile scorrere la parte visibile del foglio.

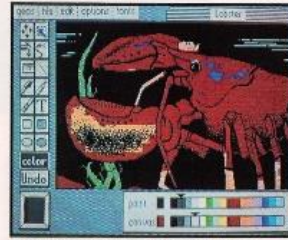
Anche la stampa di un documento creato con GeoWrite sfrutta la potenza del WYSIWYG, riproducendo su carta lo stesso risultato visualizzato sul monitor…

Sistema basato su driver

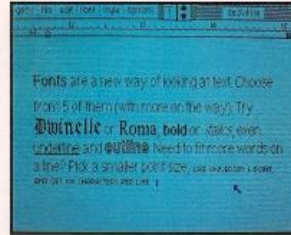
Un discorso a parte lo meritano i driver del GEOS. Il nuovo sistema operativo infatti utilizza i driver per comunicare con le periferiche di input/output. Sembra una cosa banale adesso, ma per la prima volta il sistema del C64 non si interessa della periferica, ma comunica con il driver, il quale a sua volta comunica con la periferica. Questo livello di astrazione, che ha reso famoso o famigerato Windows, sottintende il concetto di peculiarità. Finalmente era possibile utilizzare nuove periferiche sfruttandone a pieno le loro caratteristiche a patto di utilizzare il driver corretto!

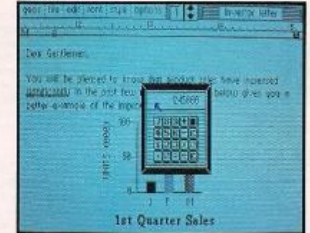
Prima di chiudere questa prima nostra digressione sul GEOS vorrei parlarvi di un'altra caratteristica fondamentale del S.O. La Berkeley Softworks ha implementato nel kernal delle primitive di accesso al disco estremamente rapide, così rapide da permettere l'utilizzo dello stesso come una memoria virtuale aggiuntiva e relativamente veloce da bypassare la limitazione fisica della scarsa dimensione della RAM del nostro amato C64. Che dire, veramente tanto di cappello a chi ha ideato questo software!

own two Machines.

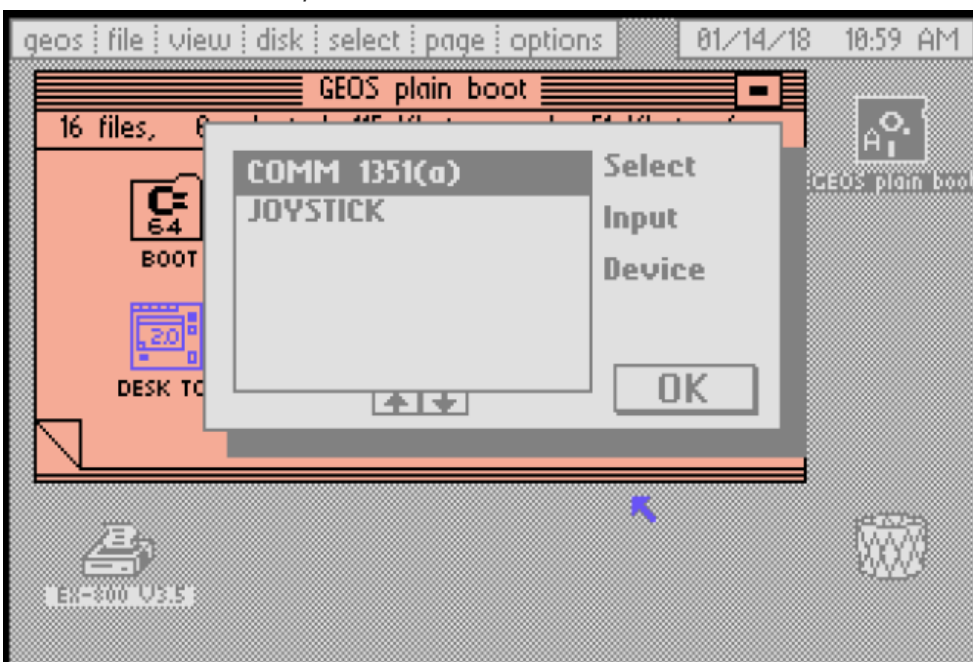
personal computer GEOS™ unlocks.

Second thoughts? Erase what you don't want. Or "UNDO" your last act. (If only life could imitate art!)

Add text if you like, in different fonts, styles or point sizes. Even change its position or layout at will.

Move or copy any part of your creation. Once done, you can include your artwork in another document—a letter home perhaps. (Won't Mother be pleased?) GEOS makes it easy.

geoWrite. An easy to use, "what you see is what you get" word processor. Create documents. Insert, copy, move or delete text as you wish. Choose from 5 different fonts in many different styles and point sizes. Preview your page exactly as it will appear off the printer. Typists will appreciate tabs, word-wrap and page breaks.

Documents may contain up to 64 pages. What's more, you can move to any page instantly. If you like, you can cut selected text from one section and move or copy it to another. Add graphics from geoPaint. It's a cinch.

Desk Accessories. Handy programs you can use while in any GEOS application. These include an alarm clock, a notepad for reminders, a four-function calculator, and photo and text albums which store pictures and phrases you may then paste into applications. The Preference Manager even lets you establish parameters for everything from mouse speed to the date and time—even background color. Civilized options, every one.

GEOS, A Whole New World for the C-64.

$59.95 TO ORDER: 800-443-0100 x234

Accanto a tanti luci pero' ci sono anche delle ombre, tutte queste funzionalita', su una macchina come il C64 si pagano in termini di velocita'. Per caricare il S.O. serve infatti un discreto lasso di tempo; tempo di attesa peraltro richiesto per qualsiasi altra operazione eseguita tramite questa innovativa interfaccia. Altra limitazione i continui cambi di disco richiesti… provate a lavorare editando una lettera con un programma che vi chiede di cambiare il disco per caricare un nuovo font o semplicemente per salvare il vostro lavoro…

Di cose da dire sul GEOS e di meraviglie da scoprire sulle sue applicazioni ce ne sono ancora tante e non stupisce che, per chi come me lo ha utilizzato da ragazzino, rimanga ancora un venerabile oggetto del desiderio.

Ritorneremo sul GEOS nei prossimi numeri, in quanto le applicazioni sviluppate per e con questo S.O. meritano di essere ricordate nel tempo perche' hanno segnato una vera e propria rivoluzione per il Commodore 64!

Per chi volesse provare ad utilizzare questo stupendo strumento, vi ricordo che il software e' liberamente scaricabile dal sito:

http://cbmfiles.com/geos/index.php

previa compilazione di un questionario in cui in pratica vi impegnate a non rivendere o ridistribuire quanto appena scaricato!

Programmazione Atari 2600 – L'hardware

di Giorgio Balestrieri

Programmare il 2600

Inizia con questo articolo una miniserie di tre orientata alla programmazione di giochi per la capostipite di tutte le console domestiche, progenitore delle superpompate Xbox e Playstation a cui siamo abituati oggi e che forse nei prossimi 10 anni spariranno, ma questa è un'altra storia (e una nostra personalissima visione del futuro). Data la natura dell'argomento, non ci è possibile mantenere un livello completamente divulgativo, i tecnicismi saranno inevitabili e una familiarità con la programmazione faciliterà molto la lettura, ma tenteremo di semplificare l'esposizione degli aspetti "per addetti ai lavori" quanto possibile.

Nata nel 1977 con il nome di Atari VCS CX2600, poi abbreviato in Atari 2600 nel 1982 e mantenuto fino al 1986, quando fu reingegnerizzata e ribattezzata Atari 2600 Jr, questa famosissima macchina da videogioco non è stata né la prima console domestica a comparire sul mercato né la prima con giochi intercambiabili della storia; il Magnavox Odyssey fu la prima in assoluto ed il Fairchild Channel F la prima ad impiegare vere cartucce (o cassette, come le si chiamava all'epoca), ma la 2600 fu quella che raggiunse il successo mondiale, tanto che ai tempi "Atari" e "videogiochi" erano sinonimi (v. articolo di Starfox Mulder su RetroMagazine n.2 per un approfondimento). Proveremo qui ad esaminare questa mitica console dal punto di vista dei programmatori, così da raccogliere un minimo di know-how necessario per sviluppare noi stessi un gioco e per poter apprezzare ancora meglio quelli che per essa furono prodotti di cui molti, per i virtuosismi tecnici richiesti, rappresentano delle vere opere d'arte informatiche, la cui realizzazione ha quasi del miracoloso.

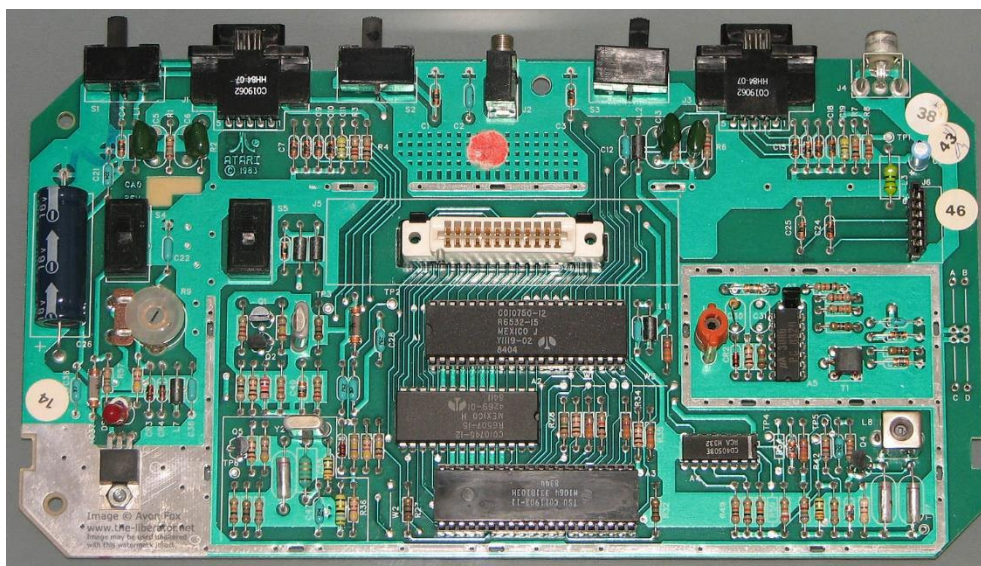



Figura 1 - Atari 2600 Jr

L'hardware del 2600

Prima di imparare a scrivere software per il 2600, è importante conoscere molto bene l'hardware di cui è composto. Questa console fu costruita con il preciso obiettivo di avere un costo di produzione più basso possibile e con in mente semplici giochi di "racchetta e palla" o di combattimento tra due giocatori, prima di essere rimpiazzata da versioni più sofisticate (quando si dice lungimiranza...), per cui è priva di componenti per la gestione di compiti in hardware come la scrittura di caratteri a video, figuriamoci se possedeva un sistema operativo o almeno un BIOS. Niente, nada, zero. Qualunque cosa si volesse far fare alla console (e intendiamo letteralmente qualunque cosa), era a carico del programmatore. Tutto ciò ha fatto guadagnare all'Atari 2600 la fama di console più difficile da programmare, roba da uomini veri, capaci di comprenderne i suoi limiti, di trasformarli in vantaggi o anche superarli, di pensare come la macchina e di scrivere software di un rigore ed una bellezza assoluti.

Per conoscere le componenti hardware di questo rude dispositivo per l'intrattenimento videoludico domestico, niente di meglio che vederle direttamente sulla scheda madre dove sono alloggiate. A questo scopo utilizzeremo la foto di Avon Fox, ottenuta dal sito the-liberator.net, riportata in **fig.1**. Si tratta della versione Junior del 2600, funzionalmente identica alle sue sorelle maggiori (per peso e volume) e più adatta per questo viaggio nell'hardware, grazie alla pulizia del design della circuiteria. Nella **fig. 2**, bordate in rosso e classificate, possiamo identificare le seguenti sezioni hardware:

Figura 2 - Atari 2600 Jr

1. Connettore cartucce a 24 pin, connesso ai bus dati ed indirizzi dei tre chip sottostanti: lo Stella, il RIOT ed il 6507.

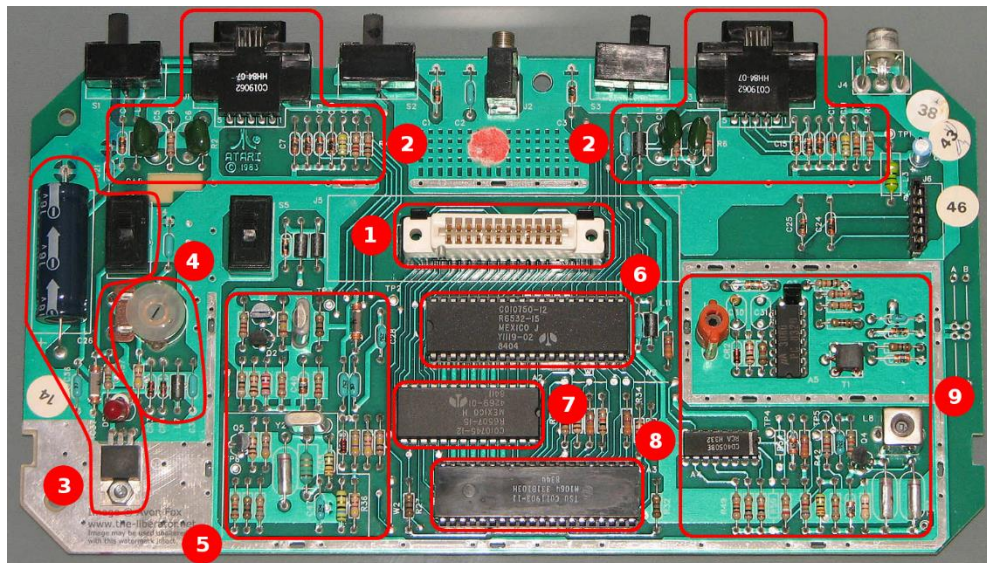
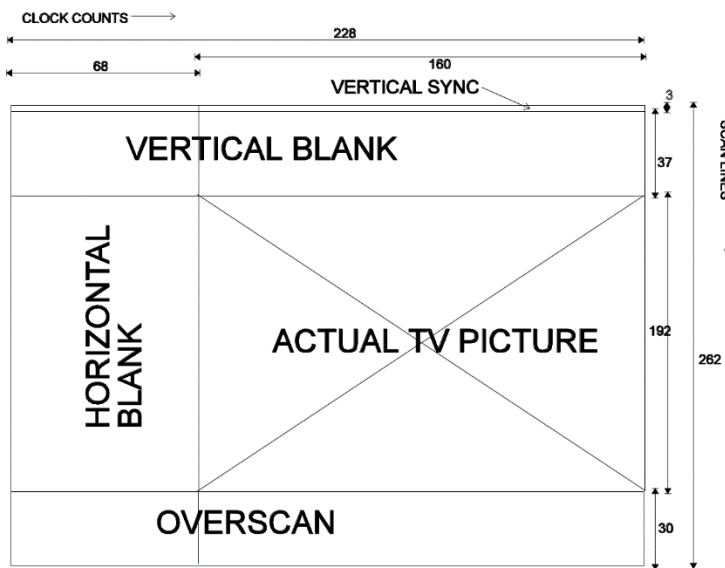



Figura 3 - Composizione frame video

2. Porte controller a 9 poli. Qui vanno collegati i joystick, i paddle ed in generale, tutte le periferiche di input. I segnali generati da queste porte sono indirizzati verso il RIOT e lo Stella. Una caratteristica poco nota è che le porte possono essere configurate anche come output, permettendo alla console di gestire eventuali dispositivi esterni.

3. Circuito di alimentazione. Riceve in input una corrente da 9 volt e la trasforma in una da 5 volt, la tensione di lavoro della scheda madre.

4. Potenziometro per la gestione della resa cromatica. Manda un segnale allo Stella che lo usa come valore base per la tonalità dei colori. Agendo sul potenziometro, si alterano i colori a video. Era utilizzato dai tecnici di laboratorio per trovare il bilanciamento cromatico ideale.

5. Circuito del clock. Genera un segnale a 3.58mhz, utilizzato dal sistema per sincronizzare le sue operazioni.

6. Chip 6532 prodotto da MOS Technology, meglio noto come RIOT, RAM I/O Timer. Il suo scopo era di leggere l'input dei controller e degli interruttori sulla console e di fornire al sistema 128 byte di memoria. Si, avete letto bene, byte, non K, mega o giga. 128 byte, 128 caratteri UTF-8 ASCII, meno di un tweet.

7. Microprocessore 6507, sempre di MOS Technology, il cervello del sistema. Girava a 1.19mhz, appena un pelino meno dei gigahertz dei moderni processori...

8. Chip audio/video TIA (Television Interface Adapter), universalmente noto come Stella.

9. Circuiteria per l'invio di suoni ed immagini elaborate da Stella al televisore.

Questa la lussuosa dotazione hardware di una delle più longeve console della storia. Per fortuna gli ingegneri della Atari conoscevano bene il proprio mestiere e concepirono il bus delle cartucce come una vera e propria porta di espansione. In questo modo fu possibile aggiungere componenti hardware al sistema, come memoria extra o circuiti audio che ampliassero le limitate capacità sonore della macchina, includendoli nella componentistica delle cartridge. Fu proprio questa caratteristica che diede alla console un ciclo vita enormemente più lungo, incredibilmente lontano dalle più rosee previsioni di ingegneri e manager della Atari.

Figura 4 - Video Olympics

Come tutte le console, anche questa è ovviamente un computer a tutti gli effetti, pur se dedicato a finalità ludiche. Curiosamente (ma non più di tanto) non era percepita come tale ma piuttosto come un sofisticato giocattolo elettronico, anche a causa della scarsa familiarità che si aveva all'epoca con i computer e l'informatica in generale. Questo accade ancora oggi per console e videogiochi, spesso etichettate troppo sbrigativamente come giocattoli ed ancora una volta il motivo è in larga parte imputabile ad una scarsa cultura informatica in generale e dei videogiochi come opere d'arte e d'ingegno in particolare. Scrivere software per questa piattaforma implica solide basi di programmazione, l'eccellente conoscenza del microprocessore 6507, soprattutto i cicli necessari per l'esecuzione di ogni istruzione ed il numero di byte richiesti (v. **riq.1**), e dello Stella. Nei prossimi paragrafi ci concentreremo nella descrizione di questi due importantissimi componenti.

In linguaggio assembly, ogni istruzione consuma byte per essere immagazzinata e cicli macchina (tempo espresso in colpi di clock) per essere eseguita. Una stessa istruzione, a seconda della modalità di accesso alla memoria, può richiedere un numero maggiore o minore di byte e di cicli macchina. Ad esempio, prendendo a riferimento l'istruzione **LDA**, che carica un valore nell'accumulatore, possiamo avere:

```
LDA #operando
```

Che carica **operando** nell'accumulatore, utilizzando l'indirizzamento immediato, richiedendo 2 byte per essere immagazzinata e 2 cicli per essere eseguita.

Invece nella forma:

```
LDA $operando
```

Utilizzata per caricare nell'accumulatore il valore contenuto nella locazione **operando** tramite indirizzamento assoluto, occorrono 3 byte e ben 4 cicli per la sua esecuzione, se **operando** è fuori dalla "zeropage" (i primi 256 byte di memoria).

Riquadro 1 – Assembly, byte e cicli

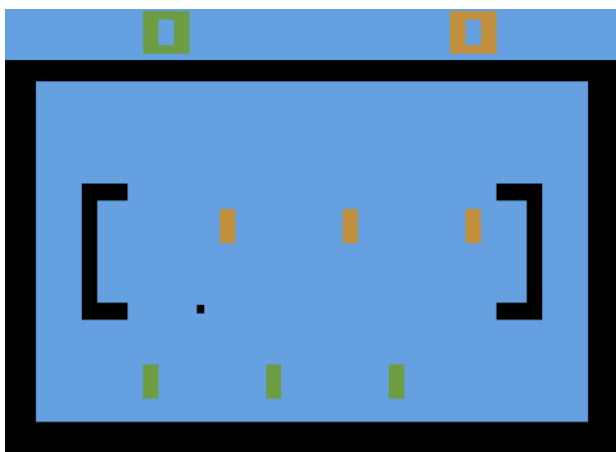



Figura 5 - Activision Pitfall!

Il MOS 6507

Il processore utilizzato dalla macchina era una versione ridotta del più famoso MOS 6502, che ha dato vita a molti dei microcomputer della 8bit generation. Per contenere i costi di produzione, vennero eliminati alcuni piedini, compresi quelli di alcune linee dati, lasciando al processore la capacità di accedere al massimo ad 8k di memoria, ridotti a 4k a causa del design del bus cartucce deciso dall'engineering Atari, che riteneva questa quantità di memoria sufficiente per la tipologia di giochi che avrebbero dovuto girare sul sistema (si pensi che il gioco Combat occupa solo 2k). I programmatori però impararono a superare questo limite ed espandere la RAM utilizzabile grazie alla tecnica del bank switching. Segmentando la memoria in blocchi da 4k (fino ad un massimo di 8) e forzando il processore ad "osservarne" uno deciso dal programmatore, si poteva ottenere l'accesso ad uno spazio di indirizzi di 32k.

Television Interface Adapter (TIA)

La gestione della generazione di immagini e suoni è affidata al TIA, meglio noto come Stella. Il chip è capace di generare immagini di 160x192 pixel sulle console NTSC e di 168x228 su quelle PAL, con un massimo di 128, 104 o 8 colori, a seconda che si utilizzi un sistema NTSC, PAL o SECAM. Può inoltre gestire due sprite (elementi grafici posizionabili arbitrariamente sullo schermo) larghi un massimo di 8 pixel, tre di un solo pixel ed un elemento per gli sfondi esteso fino a 20 pixel. Si noti che non parliamo di altezza degli oggetti disegnabili ma solo di larghezza, più avanti vedremo perché. Poiché su questa macchina il software è grandemente dipendente dal sistema NTSC o PAL su cui è destinato a girare, per semplicità qui assumeremo l'utilizzo di una console NTSC. Prima di procedere oltre, è però necessario mostrare come un televisore a tubo catodico genera le immagini e come lo Stella partecipa a questo processo.

Nei vecchi televisori, un'immagine viene composta linea per linea da un "cannone" (o pennello) che spara elettroni verso un pannello in materiale fluorescente che emette luce (in modo improprio ma comprensibile, possiamo dire che accende un pixel) nella zona colpita dagli elettroni, partendo dalla posizione in alto a sinistra e procedendo verso destra, fino al margine opposto. Il processo ricomincia identico sulla linea successiva, fino al completamento dell'immagine, in un ciclo che si ripete 60 volte al secondo (stiamo parlando di NTSC). Facendo riferimento alla **fig.3**, presa dal manuale di programmazione di Stella ad opera di Steve Wright, possiamo vedere che ogni schermata (o frame), consiste nel tracciamento di 3 linee di "vertical sync", seguite da 37 dette "vertical blank" e 30 di "overscan", il tutto temporizzato da 228 cicli di clock (color clock) durante i quali nei primi 68 battiti, detti di "horizontal blank", non c'è alcun output visibile mentre nei restanti 160 viene disegnata un'immagine a video. L'area visibile fu decisa tenendo conto delle metodologie di costruzione dei tubi catodici e dei televisori del tempo e la risoluzione di 160x192 di un quadro video in quella posizione garantiva la visione di un frame intero su qualunque TV. Notare la particolarità di avere, su uno schermo 4/3, una risoluzione verticale maggiore di quella orizzontale, l'esatto contrario di quello a cui siamo abituati. Ad ogni buon modo, tutto questo avviene in teoria ed in condizioni ideali, perché lo Stella non ha memoria video ed è compito del programmatore tracciare quanto disegnato sullo schermo modificando i registri del chip mentre il cannone elettronico spara elettroni a tutto spiano verso il pannello. In queste condizioni, la precisione del timing e della temporizzazione sono d'obbligo e molto rigide poiché qui, a differenza dei computer (retro o moderni) e delle console più evolute dove basta indicare al processore video dove sono i dati che vogliamo rappresentare, siamo noi a dover controllare l'accensione o meno di ogni pixel, riga per riga. Questo comporta che il nostro programma deve essere agganciato al ciclo del pennello che compone l'immagine e lo scorrere del tempo, o il conteggio dei colpi di clock se preferite, deve essere diviso tra quello necessario a comporre la schermata e quello impiegato per la logica di gioco e la reazione agli input del giocatore. Tenendo presente che un ciclo macchina dura tre color clock, scopriamo che abbiamo solo 76 cicli per disegnare una linea del frame, pena la comparsa di artefatti indesiderati nella nostra schermata. Questo comporta un'accorta pianificazione e una rigorosissima scelta delle istruzioni per effettuare questa delicata e critica attività e tra i programmatori divenne piuttosto comune l'espressione "racing the beam", dal momento che il software doveva letteralmente correre, cavalcando il raggio del cannone elettronico durante tutta l'esecuzione del programma, per far funzionare correttamente il gioco. Con queste informazioni, ora dovrebbe essere chiaro perché per gli elementi grafici gestiti in hardware dal TIA (sprite e fondale) non abbiamo indicato una dimensione massima in altezza: poiché Stella non ha memoria di ciò che è stato disegnato prima e dopo una riga ed è compito nostro tracciare i pixel, possiamo anche deciderne l'altezza, semplicemente obbligando Stella ad "accendere" pixel linea dopo linea. Dando una nuova occhiata alla **fig.3**, questa volta alle indicazioni del clock, con un semplice calcolo scopriamo che per 5320 cicli macchina non siamo impegnati nel disegno dello schermo e possiamo dunque impiegarli per attività come calcolare la successiva posizione degli sprite, aggiornare il punteggio, controllare eventuali impatti tra sprite e/o elementi dello sfondo ed in generale, per implementare la logica di gioco. Passati i 5320 cicli, dovremo smettere qualunque attività e ricominciare a disegnare sullo schermo. Se questo tempo non ci fosse necessario, l'unica alternativa è di rubarne altro all'attività di refresh video, rinunciando a disegnare qualche riga sopra o sotto oppure, tecnica piuttosto comune, di tracciare una linea si ed una no, riducendo di fatto la risoluzione della nostra schermata. Questo fa sì che in realtà non esista una risoluzione fissa, eccetto per quella massima raggiungibile, poiché è il programmatore a decidere quanto disegnare e quindi ad influire sulla definizione finale dell'immagine.

Figura 6 - Combat

Sprite ed elementi grafici

Dopo aver introdotto il TIA/Stella e visto come può essere utilizzato per comporre i frame video, possiamo ora analizzare in maggior dettaglio gli elementi grafici impiegati nella creazione di un videogioco. Come accennato nel precedente paragrafo, lo Stella può gestire 5 sprite, un blocco grafico per il campo da gioco ed uno sfondo, di cui è possibile cambiare solo il colore. Nessun supporto per i caratteri, il che spiega perché nei giochi, tranne che in rarissime eccezioni, il massimo impiego fatto del testo

è per il disegno dei numeri del punteggio o di quelli identificativi del gioco o della variante. Poiché il 2600 è stato progettato con lo scopo di offrire semplici giochi sulla scia di quelli esistenti, ad ogni elemento grafico è stato assegnato uno scopo ed un nome che riconduce al ruolo da ricoprire. Per questo motivo, nello Stella abbiamo:

• P0 – Player 0, lo sprite del primo giocatore.
• M0 – Missile 0, il proiettile sparabile dal primo giocatore.
• P1 – Player 1, lo sprite del secondo giocatore.
• M1 – Missile 1, il proiettile sparabile dal secondo giocatore.
• BL- Ball, lo sprite deputato a fare da pallina.
• PF – Playfield, l'elemento grafico per la composizione del campo di gioco.
• BK - Background, la parte di schermo che contiene tutti gli altri elementi grafici.

Figura 7 - Space Invaders

I primi cinque oggetti vengono chiamati anche Moveable Objects Graphics (MOG) perché sono pensati per essere posizionati e spostati in qualunque punto dello schermo, mentre il PF è progettato come elemento statico. BK, rappresentando lo sfondo e non essendo costituito da pixel si pone al di fuori di entrambe le definizioni. I MOG hanno una priorità di sovrapposizione (l'ordine in cui appaiono o sono nascosti quando si trovano l'uno sull'altro) che di default è:

- P0, M0
- P1, M1
- BL, PF
- BK

in queste condizioni, se sovrapposti P0 copre P1, entrambi coprono BL e PF, BL copre PF e tutti coprono BK. P0 e P1 coprono sempre i relativi missili. Le priorità possono essere riassegnate agendo sui registri che le gestiscono, così da avere P0 dietro agli elementi del fondale (PF) ad esempio, oppure BL che passa sopra entrambi P0 e P1.

Ad ogni elemento poteva essere assegnato un colore a piacere ma, ancora una volta, a causa di restrizioni tecniche alcuni oggetti condividevano lo stesso colore, secondo il seguente schema:

- P0, M0
- P1, M1
- PF, BL
- BK

Assegnando un colore a P0, ad esempio il verde, M0 lo eredita e si ritrova ad essere colorato in verde anch'esso. Lo stesso vale per le altre coppie indicate. La palette di colori include 128 tonalità diverse, divise in 16 colori base ed 8 livelli di luminosità in modalità NTSC. Su un sistema PAL la palette si riduce a 104 colori, mentre su SECAM i colori disponibili sono solo 8. Gli oggetti grafici sono monocromatici, cioè ad ognuno può essere associato un solo colore, ma i programmatori impararono presto a superare questo limite e creare giochi con sprite e sfondi multicolore tanto che la Imagic fece degli sprite coloratissimi un vero e proprio marchio di fabbrica. Lo Stella provvede anche ad un meccanismo per la rilevazione delle collisioni tra oggetti, attivando un bit quando almeno un pixel di un elemento grafico viene sovrapposto al pixel di un altro. In generale, ogni elemento grafico ha delle proprietà caratteristiche a seconda del ruolo assegnato e conoscerle e sfruttarle è una parte importante del bagaglio culturale di ogni buon programmatore VCS.

Gli sprite dei giocatori possono essere riflessi sull'asse orizzontale e duplicati (fino a tre copie) con una spaziatura variabile (ravvicinata o distante) tra loro, oppure è possibile raddoppiarne o quadruplicarne la larghezza. P0 e P1 hanno una dimensione orizzontale di 8 pixel, uno solo per M0, M1 e BL che possono però essere di 1, 2, 4, oppure 8 pixel orizzontali (o più precisamente, color clock). La dimensione verticale è decisa dal programmatore che indica a Stella ad ogni riga del frame se disegnare o meno lo sprite, ripetendo il pattern di 8 bit. Questa modalità era perfetta per i giochi di "palla e racchetta", come Video Olympics mostrato in fig.4, in cui P0 e P1 sono in modalità "3 copie", con un pattern di 8 pixel attivi. I programmatori si accorsero che con una programmazione accurata e sfruttando la mancanza di controlli da parte di Stella su quello che combinava lo sviluppatore, era possibile forzare l'hardware a fare ciò per cui non era stato pensato ed ottenere giochi come Pitfall! della Activision, mostrato in **fig.5**, progettato e creato dal leggendario David Crane, che per lo sviluppo di questo gioco utilizzò sofisticate tecniche di programmazione di sua invenzione. In particolare, possiamo notare l'uso di sprite multicolore e variamente modellati, ottenuti cambiando al volo il colore di P0 e P1 ed il loro pattern di 8 pixel al tracciamento di ogni riga. Ma avremo modo di parlare di questo in maggior dettaglio più avanti.

Il PF (Playfield) viene usato per disegnare gli elementi del campo da gioco. Ha un pattern largo 20 pixel che possono essere disegnati nella metà sinistra dello schermo. Per comporre la metà destra, è possibile istruire il TIA a duplicare il pattern tal quale o riflesso. Cambiando i 20 bit del pattern ad ogni riga, è possibile creare forme complesse.

Il posizionamento e lo spostamento degli oggetti è un po' più complesso. Per i cinque MOG sono disponibili dei registri TIA che permettono di scegliere in che punto dello schermo farli comparire ed in che direzione muoverli, verso destra o verso sinistra. Lo spostamento sull'asse verticale non è previsto e deve essere simulato dal programmatore agendo sui registri per nascondere e mostrare un elemento. Per il PF non è previsto lo spostamento ma, grazie al controllo completo che si ha del processo di tracciamento del quadro, lo sviluppatore può simularlo con opportune tecniche qualora gli occorra e se le sue capacità di programmazione sono eccezionalmente buone… ma d'altronde, nulla è facile su questa piattaforma. Per posizionare uno

Figura 8 - Air Sea battle

sprite sullo schermo, occorre scrivere in un registro di memoria dedicato (detto “reset position”) che ne provoca l’immediata apparizione nel punto in cui si trova il cannone elettronico. Poiché un ciclo macchina corrisponde a tre cicli del color clock, la precisione con cui possiamo piazzare il nostro sprite è di tre pixel. Per fortuna lo Stella prevede registri aggiuntivi per il posizionamento accurato degli sprite ma nonostante ciò, piazzare con precisione un oggetto a video resta un compito discretamente difficile, quasi un’arte. Lo spostamento orizzontale avviene scrivendo valori negli appositi registri durante il vertical blank. Lo spostamento può essere fino a sette pixel verso destra (+7) o otto verso sinistra (-8) a partire dall’attuale posizione dello sprite da muovere. Dopo aver scritto nel registro di spostamento di uno sprite, per ragioni tecniche non del tutto chiare, si deve evitare di scriverci di nuovo per 24 cicli macchina. Lo spostamento in verticale, come dicevamo, non è gestito dallo Stella e per essere simulato occorre nascondere e mostrare lo sprite in relazione alla riga in cui si trova ed a quella in cui lo si vuole spostare, con un attento computo del timing e delle istruzioni impiegate per manipolarlo.

Le capacità grafiche dell’Atari 2600 sono tutte qui. Nel prossimo paragrafo vedremo come alcune di esse furono impiegate nel modo canonico e come invece i programmatori seppero sfruttarle per creare effetti che lo Stella stesso non avrebbe mai immaginato di poter mostrare.

Giochi canonici e piccoli miracoli

La cartuccia che meglio si adatta a mostrare le possibilità tecniche dello Stella e del 2600 è Combat, il gioco distribuito in bundle con la console dal lancio in poi, prima di essere affiancato da PacMan nel 1982, all’uscita della versione nera del VCS denominata dai fan “Darth Vader”. Combat fu realizzato con lo scopo di mostrare le caratteristiche della macchina, così da far immaginare al pubblico il livello dei giochi che sarebbero stati prodotti. Per la sua realizzazione furono impiegate quasi tutte le capacità di Stella, ad eccezione dello sprite BL (la palla), anche se la sua funzionalità è presente in spirito nelle varianti Tank Pong dove M0 e M1 rimbalzano all’impatto con un elemento del fondale. In questo gioco sono utilizzate quasi tutte le modalità degli sprite P0 e P1 per creare carri armati, jet e biplani di dimensioni normali o doppie, in volo solitario o in formazione, attivando le funzionalità di Stella a seconda della variante di gioco e modellandoli cambiando il pattern di 8 bit ad ogni riga di schermo tracciata. Lo stesso avviene per il pattern a 20 bit di PF, utilizzato per creare campi di gioco chiusi, come nei giochi con i carri armati, o aperti per i giochi con jet e biplani e, con un moderato dinamismo contrapposto alla sua natura statica, per mostrare in alto il punteggio ed il numero della variante selezionata. Osservando (e magari, giocando) Combat, è lecito un dubbio: se Stella poteva fare solo questo, giochi come Pitfall! (**fig.5**), Space Invaders (**fig.7**) o anche Air Sea Battle (**fig.8**) come poterono essere realizzati? Di sprite ce ne sono decisamente più di 5 e Pitfall!, oltre ad impiegare MOG multicolore, sembra anche capace di disegnare linee per simulare le liane a cui il buon Harry si aggrappa per superare alcuni tipi di ostacoli. La risposta viene proprio da Air Sea Battle, o meglio da Larry Kaplan, sviluppatore Atari della prima ora, suo programmatore e creatore anche del divertente e frenetico Kaboom!

Kaplan fu tra i primi ad accorgersi che le limitate capacità della macchina potevano essere anche il suo punto di forza, poiché lasciavano al programmatore una libertà quasi assoluta nella gestione dell’hardware. Dal momento che Stella non ha memoria di ciò che fa e il compito di disegnare lo schermo è praticamente a carico del software, egli ipotizzò che manipolando i registri del TIA in maniera non convenzionale in precisi momenti del color clock, in particolare quelli del riposizionamento degli sprite dopo che i MOG erano già stati disegnati, si poteva forzare lo Stella a mostrarli di nuovo. Aveva ragione. Sincronizzando accuratamente i cicli video con quelli macchina e cambiando opportunamente i registri dello Stella, si potevano ottenere più sprite e più colori a schermo, sia in verticale, come nel caso di Air Sea Battle, che in orizzontale, come in Space Invaders, fino ad arrivare agli sprite multicolore di Pitfall, dove più tecniche vengono combinate dall’eccezionale talento di David Crane per ottenere un gioco tra i più venduti per questa piattaforma, con un “look & feel” distante anni luce da Combat e dalle idee dei progettisti del VCS. E’ interessante notare che in Pitfall! Il campo di gioco è ridotto rispetto a Combat o allo stesso Air Sea Battle. Questo perché la logica del gioco ed il livello di sincronizzazione richiesti erano talmente critici che Crane, per guadagnare tempo di calcolo (e risparmiare memoria), rinunciò ad utilizzare parte dello schermo. Ah, a proposito di Pitfall!, se vi state ancora chiedendo come fece Crane a disegnare linee animate per rappresentare una liana, la risposta è semplice (si fa per dire): calcolava il segmento che la componeva e costringeva il TIA a disegnare lo sprite della palla ad ogni riga di schermo, nella posizione in cui sarebbe dovuta essere perché l’insieme apparisse come un segmento animato. Osservando meglio la liana, i lettori più attenti potrebbero aver già notato che questa è di colore marrone tranne nel primo tratto in cui invece è verde. Ricordando che PF e BL condividono lo stesso colore, e osservando che PF è stato utilizzato sia per la chioma verde degli alberi, dove la liana è attaccata, che per i tronchi in marrone, si scopre il motivo di questo singolare effetto cromatico.

Conclusioni

Termina qui questa primo articolo sull’hardware e sulla programmazione dell’Atari 2600. Come avevamo subito messo in evidenza, scrivere software per il VCS non è un compito per programmatori novelli e quando dicevamo che è un compito “da uomini veri”, non stavamo affatto scherzando. Per nostra fortuna, l’amore dei fan per questa console ha fatto nascere più di un tool per lo sviluppo software che semplificano molto il compito a chi oggi volesse cimentarsi nella creazione di giochi. Nel prossimo numero presenteremo il bAtari Basic, un kit di sviluppo cross platform che permette di scrivere software per questa console utilizzando il linguaggio Basic.

Nel frattempo, a chi volesse approfondire meglio l’hardware o ne volesse semplicemente sapere di più sulla storia della console e di alcuni dei giochi più popolari, consigliamo le seguenti letture:

- Manuale dello Stella di Steve Wright, ricostruito da Charles Sinnett, recuperabile da atarihq.com o, per chi preferisce gli originali, su archive.org.

- Specifiche dell’Atari 2600, dal sito problemkaputt.de ad opera del singolare e talentuoso sviluppatore Martin Korth.

- L’eccellente libro “Racing the beam”, di Nick Montfort e Ian Bogost.

Figura 9 - Solaris

Generare toni DTMF con il SID del C64

di Marco Pistorio

Cosa sono i toni DTMF?

Spieghiamo innanzitutto cosa sono i toni DTMF. DTMF è un acronimo che sta per Dual-Tone Multi-Frequency. E' un sistema di codifica utilizzato nella telefonia. In altre parole, ad ogni numero digitato sulla tastiera dell'apparato telefonico, corrisponde un tono DTMF specifico, ben preciso.

Prima dell'introduzione del DTMF (noto anche come sistema multifrequenza) era utilizzato il sistema ad impulsi, che veniva anche denominato sistema "decadico". Sull'apparecchio telefonico era presente un disco, il cosiddetto "disco combinatore", con una serie di fori corrispondenti alle diverse cifre. Il disco veniva azionato inserendo un dito nel foro corrispondente alla cifra da comporre e ruotandolo fino ad un fermo. Una volta rilasciato ritornava alla posizione iniziale e nel contempo agiva su un interruttore che interrompeva la linea telefonica un numero di volte corrispondente alla cifra composta. Molti di Voi che hanno una età anagrafica che va dai 30 anni in su probabilmente ricorderanno i telefoni con il disco combinatore che venivano installati dalla SIP, la principale azienda di telecomunicazioni italiana per un certo periodo, attiva dal 1964 al 1994, poi trasformata in Telecom Italia spa.

Tali apparecchi si intravedono comunque spesso sia nel piccolo che nel grande schermo, quando vengono proposte pellicole girate in quegli anni oppure che rievocano fatti e/o personaggi collocati grossomodo nello stesso periodo temporale. I telefoni DTMF in Italia divennero sempre più diffusi a cavallo tra la fine degli anni '80 ed i primi anni '90 per poi diventare lo standard. Vediamo adesso di comprendere alcuni aspetti di questa tecnica.

Come si generano i toni DTMF

E' necessario comprendere che un tono DTMF è un suono generato contemporaneamente su due frequenze ben precise, con caratteristiche studiate in maniera accurata. Le frequenze sono state scelte in modo che le armoniche e le intermodulazioni non generino segnali rilevanti. Nessuna frequenza è un multiplo intero di un'altra e la differenza e la somma tra due frequenze non corrisponde ad alcun tono.

I toni DTMF devono viaggiare dall'apparato telefonico alla centrale telefonica. Ciò comporta che i toni devono essere generati all'interno della banda di frequenze udibili, ovvero lo stesso intervallo di frequenze che ci permette di conversare con l'interlocutore che risponde alla nostra chiamata. Tali toni serviranno per selezionare il numero verso il quale si intende effettuare la chiamata. Inizialmente gli apparecchi telefonici (comunemente detti "telefoni", si, proprio loro...) non prevedevano filtri per nascondere i toni DTMF che venivano composti.

Oggi invece è consueto l'esatto contrario: quando componiamo le varie cifre del numero che intendiamo chiamare, il tono relativo a ciascuna cifra composta viene filtrato in maniera tale da rendere i toni non udibili. Filtri o no, la logica del sistema DTMF comunque resta la stessa. Segue uno schema che illustra la relazione tra cifre ed i relativi toni corrispondenti:

1	2	3	A	**697 Hz**
4	5	6	B	**770 Hz**
7	8	9	C	**852 Hz**
*	0	#	D	**941 Hz**
1209 Hz	**1336 Hz**	**1477 Hz**	**1633 Hz**	

Da tale tabella si intuisce che, per esempio, la cifra "1" è codificata mediante due segnali, uno a frequenza 697 Hz e l'altro a frequenza 1209 Hz.

Le lettere A,B,C e D non sono implementate nelle tastiere dei telefoni, ma sono usate per gestire funzioni interne della rete telefonica. Sono usate anche in applicazioni non telefoniche come il controllo dei ripetitori radioamatoriali, ad esempio.

In generale i toni DTMF vengono adoperati per effettuare una chiamata telefonica, ma anche per delle semplici basi musicali di risponditori automatici e nell'uso di canali trasmissivi diversi dal classico doppino in rame come ponti a microonde e via satellite.

Spero di aver detto tutto il necessario.

Per chi volesse approfondire comunque, fornisco il link della pagina di wikipedia relativa all'argomento:

https://it.wikipedia.org/wiki/Dual-tone_multi-frequency

Usare il C64 per generare i toni DTMF

Dopo la parte teorica (spero non troppo barbosa) passiamo alla pratica e vediamo come sfruttare il C64 per generare toni DTMF.

Vi anticipo subito che l'uso del C64, e del SID in particolare, risulta in questo frangente molto comodo. Perchè? Perchè i toni DTMF vengono generati su due frequenze, contemporaneamente.

Quale caso migliore per sfruttare due dei tre canali a disposizione del chip audio del C64, il SID?

Ecco quindi il listato, rigorosamente in BASIC, che Vi permetterà di raggiungere questo scopo:

```
0 POKE53280,0:
POKE53281,0:POKE646,5
1 PRINT CHR$(147);
2 GOSUB 1000: REM IMPOSTA
TABELLA FREQUENZE
5 :
6 READ N:IF N<>-1 THEN PRINT
N;:GOSUB 2000:GOTO 10
8 GOTO 99
9 :
10 REM GENERA TONI DTMF
11 SI=54272
12 POKESI+24,15: REM VOLUME
MAX
```

```
13 POKESI+5,0:POKESI+6,240: REM
ATTACK-DECAY-SUSTAIN-RELEASE
VOCE1
14 POKESI+12,0:POKESI+13,240: REM
ATTACK-DECAY-SUSTAIN-RELEASE
VOCE2
15 POKESI+1,A:POKESI,B: REM
GENERA TONO ALFA
16 POKESI+8,C:POKESI+7,D: REM
GENERA TONO BETA
17 POKESI+4,17: REM FORMA
D'ONDA VOCE1
18 POKESI+11,17: REM FORMA
D'ONDA VOCE2
19 FORL=0TO300:NEXT:REM PAUSA
20 :REM SPEGNI TONI
80
POKESI+4,0:POKESI+5,0:POKESI+6,0
81
POKESI+11,0:POKESI+12,0:POKESI+1
3,0
85 FOR L=0 TO 50:NEXT:REM PAUSA
BREVE
86 GOTO 6
87 :
99 PRINT:END
998 :
999 REM TABELLA FREQUENZE
1000 H(1)=46:L(1)=93 :REM 697 HZ
1001 H(2)=51:L(2)=57 :REM 770 HZ
1002 H(3)=56:L(3)=173 :REM 852 HZ
1003 H(4)=62:L(4)=153 :REM 941 HZ
1004 H(5)=80:L(5)=109 :REM 1209 HZ
1005 H(6)=88:L(6)=224 :REM 1336 HZ
1006 H(7)=98:L(7)=65 :REM 1477 HZ
1007 RETURN
1999 :
2000 REM CALCOLO PARAMETRI
2001 IF N=1 THEN
A=H(1):B=L(1):C=H(5):D=L(5)
2002 IF N=2 THEN
A=H(1):B=L(1):C=H(6):D=L(6)
2003 IF N=3 THEN
A=H(1):B=L(1):C=H(7):D=L(7)
2004 IF N=4 THEN
A=H(2):B=L(2):C=H(5):D=L(5)
2005 IF N=5 THEN
A=H(2):B=L(2):C=H(6):D=L(6)
2006 IF N=6 THEN
A=H(2):B=L(2):C=H(7):D=L(7)
2007 IF N=7 THEN
A=H(3):B=L(3):C=H(5):D=L(5)
2008 IF N=8 THEN
A=H(3):B=L(3):C=H(6):D=L(6)
2009 IF N=9 THEN
A=H(3):B=L(3):C=H(7):D=L(7)
2010 IF N=0 THEN
A=H(4):B=L(4):C=H(6):D=L(6)
2011 RETURN
2999 :
3000 DATA 8,0,0,0,5,5,0,7,7,-1
```

Il listato credo sia già sufficientemente commentato ma spenderò comunque due parole circa il suo funzionamento.

Il programma legge le cifre da comporre dalla riga DATA numero 3000. Ciascuna cifra viene letta, una per volta. Se la cifra letta è -1, significa che non ci sono più altre cifre da elaborare ed il programma termina. Per tutti gli altri valori di cifra, da 0 a 9, viene richiamata la routine a partire da riga 2000 che, in base alla cifra da comporre, imposta opportunamente i valori contenuti nelle variabili A,B,C e D. Tali valori verranno adoperati per la generazione dei due toni corrispondenti alla cifra. Vi ricordo che ciascun valore di frequenza viene ottenuto mediante una coppia di valori, che corrispondono alla parte bassa ed a quella alta della frequenza audio che si intende generare.

Per comprendere meglio tali valori può risultare utile consultare la tabella che si trova nell'appendice M del manuale d'uso del C64. Partendo dai valori ivi presenti, collegati alle relative frequenze, è possibile ricavare empiricamente una funzione che permetta di calcolare, data una frequenza X le due componenti LO-HI. Una forma di tale espressione empirica potrebbe essere la seguente:

(C64 PAL) x = f * (18*2^24)/17734475 con f,(Frequenza) 0 - 3848 Hz

(C64 NTSC) x = f * (14*2^24)/14318182 con f,(Frequenza) 0 - 3995 Hz

Dividendo il numero x per 256 e considerando la parte intera del risultato di tale divisione si ottiene la parte hi della coppia. Sottraendo al numero x il valore ottenuto da 256*hi si ottiene la restante parte lo. Per ottenere le coppie di numeri corrispondenti alle frequenze richieste in maniera il più possibile precisa, ho adoperato inoltre un algoritmo di calcolo numerico denominato "best-fit".

Attenzione: I valori presenti nel listato BASIC sono validi per un C64 che funzioni in modalità PAL.

Avvicinando una cornetta di un telefono (oppure un cellulare in modalità di attesa numero da chiamare) in prossimità del diffusore dell'audio generato dal C64 (che spesso è l'altoparlante all'interno del Monitor/TV), verrà composto il numero impostato tramite la generazione dei toni corrispondenti.

Ah, il numero 800-055077, per chi non lo sapesse, è quello di "SOS Gabibbo". Non ditemi poi che non ve lo avevo detto... :)

Curiosità

Una nota ulteriore relativa ai toni DTMF. Come Vi accennavo, è possibile definire delle semplici melodie oltrechè comporre meramente dei numeri telefonici.

Per chi fosse interessato, ecco una sequenza di cifre che Vi suggerisco di provare:

1231 1231 369 369 908521 908521 021 021 ;

Se la memoria non mi inganna, dovrebbe corrispondere al ritornello di "Frà Martino" :)

Altra sequenza graziosa:

133 3213 3656963 365 587 745 521 133 369 000 ;

che riproduce in maniera abbastanza similare la canzoncina di "Popeye". Queste due sequenze le ho ricavate da un taccuino che conservo gelosamente da oltre vent'anni!

E' sufficiente cambiare il contenuto della riga DATA digitandovi il contenuto della prima oppure della seconda sequenza per ottenere l'uno o l'altro risultato, magari su più linee DATA di seguito.

Spero che l'articolo abbia stimolato la Vostra curiosità, che Vi sia piaciuto e che Vi abbia fatto tornare con la memoria un po' indietro negli anni, perché no? Alla prossima.

Bibliografia

https://it.wikipedia.org/wiki/Dual-tone_multi-frequency

https://www.c64-wiki.com/wiki/SID

http://www.oxyron.de/html/registers_sid.html

Max Machine e la cartuccia MultiMAX

di Codingkoala

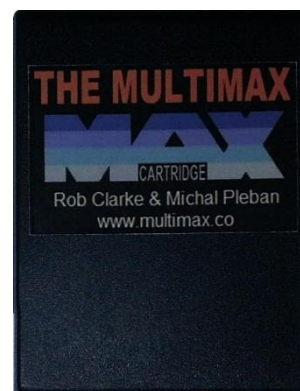
Sembra strano ma gia' nel lontano 1982 la guerra fra console era una realta' e la Commodore decise di lanciare la sua sfida alle emergenti console giapponesi proponendo la sua MAX Machine, chiamata in Germania VC-10 e negli USA Ultimax. La MAX Machine era sostanzialmente una versione depotenziata del C64, stessa architettura basata su 6510 a 1MHz, solo 2K di RAM (e 0.5KB di Color RAM), VIC-II (MOS 6566) e SID 6581, niente porta seriale quindi niente disk drive o stampante, niente user port, nessuna ROM e nessuna cartuccia inclusa nel pacchetto di vendita.

La MAX Machine fu prodotta in circa 50.000 unita' ma viste le caratteristiche limitate e la enorme popolarita' e capacita' del VIC-20 fu presto messa fuori produzione. L'assenza di ROM fa si che senza una cartuccia inserita la MAX Machine sia praticamente inutilizzabile generando nient'altro che uno schermo nero, per inciso la MAX Machine essendo nata esclusivamente per il mercato Giapponese ha una uscita video in formato NTSC-J, ovvero la particolare variante NTSC diffusa fino al 2011 in Giappone ma ormai sparita a causa dallo switch alle trasmissioni in digitale, che e' ibrido tra l'NTSC americano e il PAL europeo. Da notare che la MAX Machine fu presentata nel noto modello con tastiera a membrana ma anche in un modello prototipo con scheda madre alloggiata in un case simile a quello del C64 ma senza tastiera con un uno slot per cartucce nella parte superiore ... in definitiva un C64GS con 8 anni di anticipo! La cartuccia piu' semplice creata per MAX Machine e' il MINI Basic, una versione ridotta del BASIC V2 con solo 510 byte liberi, mentre la cartuccia MAX BASIC implementa il BASIC V2 completo con 2047 byte liberi grazie ad una RAM da 2K inclusa nella cartuccia.

La HAL Laboratory, software house Giapponese nota per diversi giochi per VIC-20, aveva convertito alcuni dei suoi titoli per MAX Machine e C64. Per MAX Machine sono state in definitiva prodotte appena 24 cartucce (vedi box a lato).

La miglior cartuccia per un possessore di MAX Machine e' la MultiMAX, una cartuccia che riunisce tutte le cartucce prodotte per MAX nelle loro diverse versioni oltre ad includere 2K di RAM per il MAX Basic e un comodo menu, che permette di scegliere il banco della eprom da lanciare, e un tasto di reset. Avendo una MultiMAX non e' necessario inserire altre cartucce e quindi preservare la posta sul retro della MAX Machine.

L'architettura della cartuccia e' piuttosto semplice, usa una Eprom 27c801 da 1MB e un chip RAM 6116 da 2K. La Eprom e' suddivisa in 64 banchi da 16K, ogni banco contiene le immagini da 8K dei due chip delle cartucce per MAX. Un latch resettabile 74LS273 si occupa di selezionare il banco corrente e il reset permette di tornare sempre al banco 0 (Menu) al reset, il latch e' mappato in memoria e quindi il banco e' selezionabile da software (dal menu). Il settimo bit del latch e' usato come un segnale particolare che rimuove dalla mappa di memoria il latch stesso e seleziona il 6116, quindi selezionando un banco on il settimo bit attivo si abilita la RAM aggiuntiva di 2K. La cartuccia ha poi due chip 74LS08 e 74LS27 che si occupano dei segnali di controllo per la Eprom, la RAM e il latch.

Di recente diverse MAX Machine sono arrivate in Italia ed alcuni miei amici hanno avuto il piacere di venirne in possesso, per questa ragione ho deciso di realizzare la MultiMAX Replica ovvero una replica fedele della MultiMAX. Tale scheda ovviamente grazie alla modalita' Ultimax gira perfettamente anche su C64.

LE 24 CARTUCCE DELLA MAX MACHINE

- AVENGER
- BILIARDS
- BOWLING
- CLOWNS
- GORF
- JUPITER LANDER
- KICKMAN
- LE MANS
- MAX BASIC
- MINI BASIC
- MOLE ATTACK
- MONEY WARS
- MUSIC COMPOSER
- MUSIC MACHINE
- OMEGA RACE
- PINBALL
- RADAR RAT RACE
- ROAD RACE
- SEA WOLF
- SLALOM
- SPEED/BINGO MATH
- SUPER ALIEN
- VISI SOLAR SYSTEM
- WIZARD OF WOR

Console 8bit + Guest: **Mattel Intellivision**

di Starfox Mulder

Bentornati alla rubrica che si occupa delle console ad 8 bit. Questa volta vi presento un pezzo di cuore: il mio amato Inty. Prima ancora di aver appreso come afferrare un controller già ne possedevo uno, dato che mio padre fu tra i rappresentanti italiani del prodotto nei primi anni 80 ed io iniziai ad amare i videogiochi guardandoli giocare da altri. Nostalgia a parte, qui ci vuole una precisazione perché la rubrica si chiama "Console 8bit + Guest" ed infatti eccoci già a parlare del "Guest". L'Intellivision non aveva un cuore ad 8 bit ma fu anzi la prima console a 16 bit della storia. Con il suo processore CP1600 della General Instrument Mattel entrò a gamba tesa nel mondo dei videogiochi, fino ad allora dominato unicamente da Atari dato lo scarso successo che le precedenti concorrenti avevano ottenuto, puntando su pubblicità comparative che sottolineassero la maggiori prestazioni ed il comparto grafico superiore in tutto. Sappiamo oggi quanto le console wars siano mere strategie di marketing e nulla più ma per l'epoca la scelta fu vincente e la console Mattel seppe ritagliarsi una bella nicchia di mercato in America come nel resto del mondo.

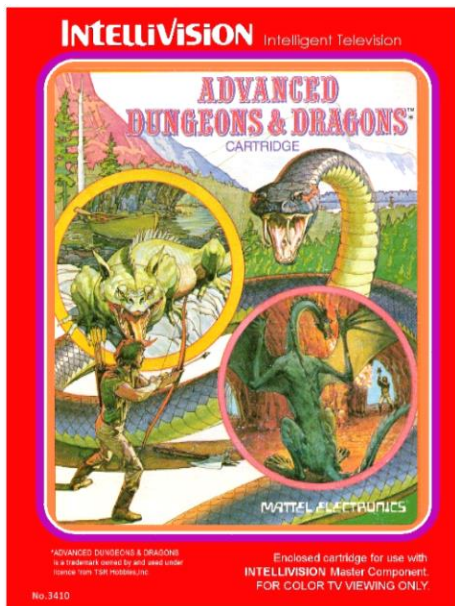
Sì, ma i giochi? Centoventicinque in tutto, tenendo conto del periodo in cui la console venne commercializzata ed ignorando per ora gli homebrews successivi. Il principale problema di Mattel fu da subito che non possedeva i diritti delle conversioni da arcade, dal momento che Atari se li era accaparrati pressoché tutti. Space Invaders? Certo, ma su Inty dovette chiamarsi Space Armada. Phoenix? Ci si dovette accontentare del clone Demon Attack e così via. Questa mancanza di titoli di richiamo venne però compensata con un impegno serratissimo dalla stessa casa produtrice nel far sì che venissero sviluppati titoli originali ed esclusivi. Ecco quindi nascere i giochi del mitico film Tron, le prime licenze TSR su Dungeons & Dragons ed esperimenti coraggiosi come Utopia, progenitore di Civilization, o Microsurgeon, il primo Health Educational Game. Il divertimento non mancava ed essendo figlia di un epoca in cui si amava sperimentare non mancarono neppure gli add-on coraggiosi e un sistema di comandi ancora oggi ostico da padroneggiare. I due controller inglobati alla console presentavano un disco direzionale a 16 direzioni, un tastierino numerico su cui inserire delle speciali cover che ogni gioco presentava in esclusiva e dulcis in fundo dei grilletti laterali adattissimi agli sparatutto pur provocanti crampi ai pollici di inenarrabile dolore. Le periferiche videro l'intellivision trasformarsi in una sorta di personal computer grazie alla tastiera, alla sintesi vocale e ad un rudimentale sistema di scaricamento giochi tramite linea telefonica (solo negli states e col difetto di formattare ogni volta allo spegnimento della console). Un avversario di tutto rispetto per la potentissima Atari sebbene a sua volta non seppe far fronte alla crisi del 83. Quando il mondo dei videogiochi crollò, Intellivision si ripropose in successive versioni più economiche fino a giungere al progetto Intellivion Lives per recenti console. Vere e proprie collection dei titoli storici riproposti in una hub moderna ma senza il fascino della sincronizzazione canale e dei pad scomodissimi sappiamo benissimo non essere la stessa cosa.

Alla prossima console!

CARATTERISTICHE TECNICHE

Produttore	Mattel
Tipo	Console 16 bit
Generazione	Seconda
In vendita	Maggio 1979
Dismissione	1990
Supporto	Cartuccia
Unita' vendute	Oltre 3 milioni

FONTE: HTTPS://EN.WIKIPEDIA.ORG/WIKI/INTELLIVISION

AD&D CLOUDY MOUNTAIN

Le copertine di una volta...

...erano dei veri e propri capolavori, una gioia per gli occhi. Alzi la mano chi non ha mai comprato un gioco attratto dalla sua confezione.

Effetti sonori icredibili

Ogni mostro ha il suo effetto sonoro che ne preannuncia l'arrivo e spesso ci permetterà di attaccarlo alla cieca prima ancora di essere entrati nella stanza in cui dimora, unico modo per sopravvivere alla modalità very hard.

GIUDIZIO SUL GIOCO

GIOCABILITA'

90%

Il pad dell'Intellivision non è mai stato per tutti ma se saprete dominarne le caratteristiche qui vi ci approccerete al meglio delle sue possibilità, in un esperienza immediata e sempre divertente.

LONGEVITA'

95%

Quattro livelli di difficoltà, dungeon e mappe di gioco generati randomicamente uniti ad una capacità di coinvolgimento unica.

Advanced Dungeons & Dragons Cloudy Mountain

Mattel Elettronics - Anno 1982 - Piattaforma Mattel Intellivision

Nello stesso numero in cui vi presento la console della mia infanzia non posso esimermi dal recensirvi il suo gioco più iconico. So quanto questa definizione sia soggettiva ma credetemi se vi dico che dopo averlo provato come si deve ne resterete rapiti. Il primo videogioco della storia a possedere la licenza ufficiale TSR (la casa editrice del gioco di ruolo cartaceo) riprendeva davvero poco del gdr d'ispirazione, ma possedeva un gameplay eccezionale.

I giocatori si trovano calati nel ruolo di un elfo armato di arco e (poche) frecce. Tre sole vite ed una mappa intera da esplorare per giungere fino alla montagna del fato dove dovremo affrontare i due temibili draghi a due teste per potergli sottrarre i pezzi della corona e ricongiungerli al fine di terminare il tutto. Nel mentre l'esplorazione la farà da padrona.

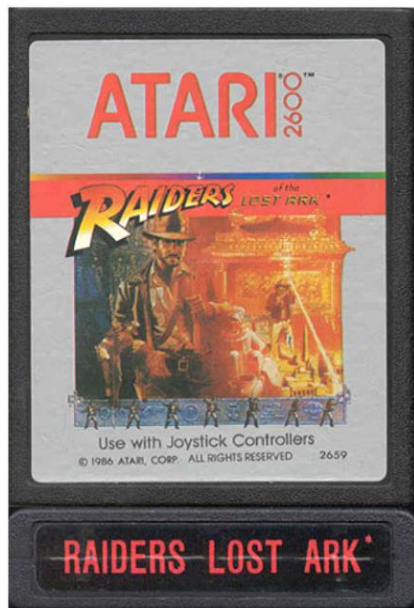
Ogni volta che accenderemo la console e sceglieremo una delle quattro difficoltà infatti, il gioco genererà casualmente la mappa del mondo, così come farà per ogni singolo livello in cui entreremo. Gestione procedurale delle mappe quindi, con nemici casuali ed oggetti piazzati in modo del tutto randomico (Perche' non casuale? NdFF) all'interno dei vari dungeon.

Eppure una logica c'è. Avvicinandoci ad un dungeon ci verrà mostrato il suo "colore" da cui potremo dedurre sia la tipologia di creature che lo abitano, sia l'equipaggiamento recuperabile al suo interno. Le montagne bianche saranno fonte di munizioni. Utili frecce da inserire nella nostra faretra per affrontare i numerosi nemici. Con il tasto 0 potremo richiamare un conteggio delle stesse ma non aspettatevi un numero a schermo: verranno conteggiate da un effetto sonoro quindi state all'orecchio e contate velocemente. Le montagne blu invece ci permetteranno di recuperare un'imbarcazione con cui attraversare i fiumi, quelle rosse ci forniranno di un ascia con cui farci largo nei fitti boschi ed infine le montagne viola ci daranno l'accesso alla chiave con cui aprire i cancelli antistanti la montagna del fato.

Se l'utilizzo del tastierino numerico ci permetterà di muoverci sulla mappa del mondo, sarà invece il disco direzionale a orientare il nostro alterego una volta nei dungeon, pur non lasciando la mano dal tastierino che d'un tratto diverrà il mezzo per direzionare il nostro arco. Come nel noto Robotron 2084 avremo quindi una gestione doppia di direzione e sparo ma stavolta con lo stesso pad e con l'aggiunta di un tasto per uscire dai dungeon (una volta trovata la scala che ci condurrà fuori), uno per raccogliere oggetti ed uno per conteggiare le freccie.

Di Dungeons and Dragons restano i labirinti, i mostri, il senso di avventura ed il contesto fantasy ma si perde tutta la componente evolutiva, poi recuperata nel suo seguito diretto Treasure of Tarmin che per contro perde l'immediatezza e la giocabilità di questo titolo. Potrei incensarlo a lungo ma vi basti sapere che è tra i pochissimi giochi che se inserisco nella console mi trovo inevitabilmente a giocare per ore. Ogni singola volta. Magari non avremo più lo spirito del me stesso bambino che si terrorizzava quando giunto nei pressi della stanza di un mostro se ne sentiva i ruggiti o la capacità di immaginarsi più di quel che la console sapeva rendere graficamente, eppure la qualità non è calata col tempo e questo non si può certo dire di altri contemporanei ben più blasonati.

di Starfox Mulder

RAIDERS OF THE LOST ARK

Inizio del gioco

L'arca ci viene mostrata nella schermata di inizio del gioco, con la colonna segnapunti alla sua estensione massima. Rapidamente questa si abbassa fino a sparire (insieme all'arca), lasciando Indy alle prese con la sua avventura per ritrovarla.

GIUDIZIO SUL GIOCO

GIOCABILITA'

85%

Il sistema di controllo a due joystick vi farà dannare non poco, ma con la pratica imparerete ad anticipare la selezione dell'oggetto che vi occorre ed a farlo nei momenti di calma.

LONGEVITA'

90%

Arrivare alla soluzione dell'avventura vi richiederà un gran numero di partite, e una volta finito, avrete voglia di rigiocarlo fino a scoprirne tutti i suoi segreti e raggiungere il punteggio massimo. Dopodichè, come per tutte le avventure, la voglia di giocarlo ancora crolla praticamente a zero, ma è il destino di tutti i giochi di questo tipo.

Raiders of The Lost Ark

Atari - Anno 1982 - Piattaforma Atari 2600

Indy alla conquista dei Videogiochi

A poco più di un anno dalla grande crisi (americana) dei videogiochi, la direzione Warner di Atari capitanata da Kassar decise che dopo Superman, era il momento di un altro un tie-in cinematografico. L'enorme successo del film e la familiarità di Warner con questo tipo di prodotto, radicarono velocemente nel top management Atari la convinzione che l'idea di un gioco con lo stesso titolo della pellicola ed Indiana Jones stampato bello grande sulla scatola era dannatamente buona e avrebbe portato nelle casse dell'azienda vagonate di dollari. Affidarono il progetto ad Howard Scott Warshaw, che aveva lasciato il rigido mondo della HP per quello lassista e godereccio di Atari e vantava al suo attivo il campione di vendite Yar's Revenge, ancora oggi considerato uno dei migliori giochi per VCS. Warshaw incontrò Spielberg per presentargli la sua visione della trasposizione videoludica de "I predatori dell'arca perduta", gli mostrò Yar's Revenge e gli disse di voler creare un gioco di avventura dinamica sulla base di Adventure (sempre della Atari) ma con più enigmi da risolvere e una componente di azione più marcata, che si sarebbe svolta nelle location centrali del film, quelle cioè in Egitto. Howard riuscì a convincere Spielberg della validità delle sue idee ed ottenne la sua approvazione come sviluppatore del progetto. Il regista gradì molto anche Yar's Revenge ma conoscendo la passione di Spielberg per la fantascienza e la qualità del gioco, era difficile andasse diversamente. Warshaw dal canto suo fu entusiasta del lavoro affidatogli; fare il tie-in di un film lo stimolava molto più della conversione di un arcade, al punto che prese a girare in ufficio indossando un cappellaccio come quello di Indy ed una vera frusta di pelle lunga un metro e mezzo, che si divertiva a far schioccare spaventando a morte chiunque avesse la sventura di trovarselo alle spalle (si, il gruppo di sviluppo Atari era composto da una banda di geek pazzi furiosi). Lo sviluppo iniziò nell'estate del 1981 ed il gioco fu messo in commercio nel novembre del 1982. Fu molto apprezzato da critica e pubblico, donando a Warshaw il suo secondo successo.

Il gioco

Partendo dalla struttura di Adventure di Warren Robinett, Warshaw ebbe l'idea di sviluppare un gioco dove alla meccanica di esplorazione e risoluzione degli enigmi fosse affiancata una robusta dinamica di azione. Raiders of the lost ark, contrariamente al gioco di Robinett, disponeva di un vero inventario il cui uso, invece di una schermata dedicata, avveniva tramite l'utilizzo del secondo joystick mentre l'azione era in corso, così da rendere il tutto più frenetico e difficile. Purtroppo, a causa del ritmo di gioco e della poca praticità di riuscire a manovrare due controller già non comodissimi di loro (avevano la tendenza a provocare crampi fenomenali), perdere una vita o un oggetto perché uno dei joystick scivolava via o si premeva il tasto sbagliato era abbastanza frequente. Questo, in combinazione con il fatto che il gioco era piuttosto cattivello per cui perdere un oggetto poteva impedire di completare l'avventura, era fonte di forte frustrazione e causa della violazione dei divieti di molte religioni circa l'invocazione per futili motivi delle divinità più disparate...

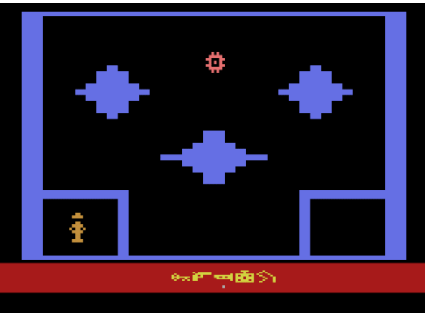
Il Cairo e l'arca perduta

Nel gioco come nel film, dovremo girovagare per il Cairo e dintorni al comando dell'archeologo più famoso del mondo alla ricerca dell'Arca dell'Alleanza. La mappa di gioco si sviluppa su tredici locazioni (dette "stanze"), di cui solo dieci vengono menzionate nel manuale (anche se c'è un accenno alla stanza delle mappe), lasciando al giocatore il compito di scoprire le altre. L'accesso alle varie stanze non è sempre visibile e per guadagnarlo, a volte dovremo esplorare con cura i dettagli di ogni locazione ed altre utilizzare l'oggetto giusto. L'Egitto pullula di pericoli e ladri, serpenti, mosche tsetse, vagabondi impazziti e criminali in genere saranno fin troppo solerti nel cercare di complicarvi la vita, rubandovi oggetti, rallentandovi o dandovi la morte. Anche la natura è contro di noi e impantanarsi nelle paludi o cadere da una mesa è fin troppo facile per il giocatore alle prime armi che non ha ancora sufficiente conoscenza del territorio. Per fortuna ci sono due mercati, in cui sono presenti personaggi che se non proprio amici, almeno non ce l'hanno con noi e sono disposti ad aiutarci in cambio di soldi (suggerimento: un flauto è un'ottima difesa contro i serpenti). Nei mercati si trovano anche delle ceste che possono fornire oggetti utili; meglio controllarle tutte, magari più volte. Altri oggetti possono essere trovati nelle stanze segrete, anche se non sempre sono visibili o necessari a completare il gioco. Quali ci saranno utili e quali no, sta a noi scoprirlo. E' presente anche un sistema di

Il mercato

Qui possiamo trovare molti oggetti utili al nostro scopo, ma attenzione ai serpenti!

Stanza d'entrata

Indy inizia qui la sua avventura. Come è stato possibile combinare quel macello sulla parete di destra?

Stanza della luce splendente

Imprigionati in un'angusta cella, alla ricerca di un modo per evadere.

La valle del veleno

Una locazione decisamente letale, dove ogni cosa è contro di noi e trama per impedirci di uscirne vivi.

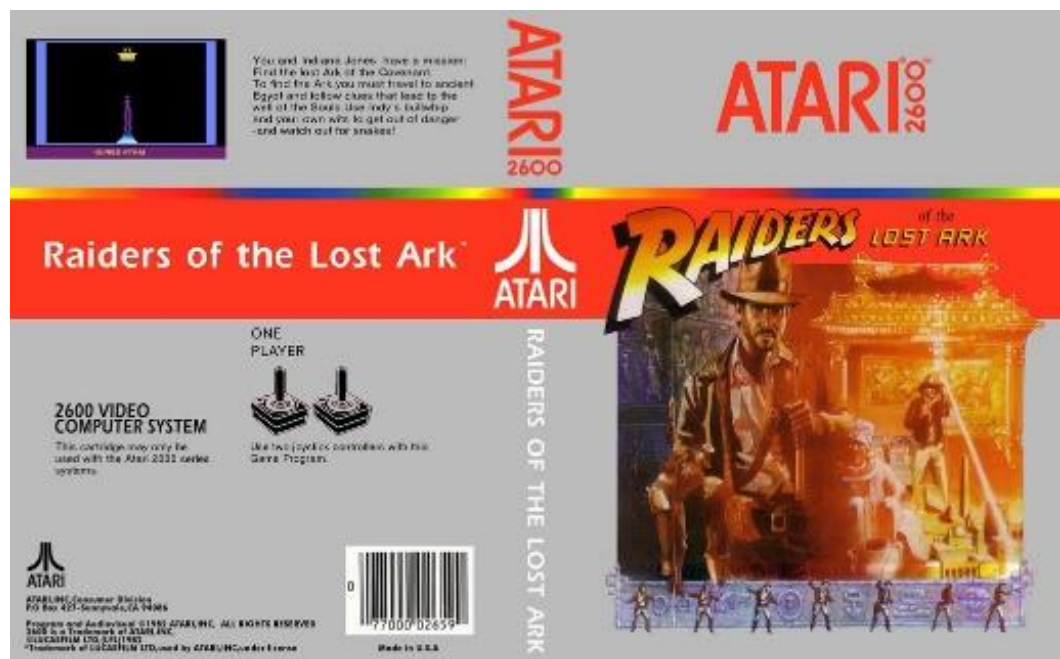
punteggio, rappresentato dall'altezza di una colonna su cui viene posto Indy una volta trovata l'arca. Più alta sarà la colonna, meglio avremo giocato e se saremo riusciti a raggiungere il punteggio massimo, verremo gratificati da una iscrizione speciale, il cui contenuto non è svelato dal manuale. Se non resistete alla curiosità, con una rapida ricerca sul web scoprirete presto di cosa si tratta.

Come da tradizione, anche qui Warshaw ha lasciato un riferimento nascosto a se stesso, stavolta al suo primo gioco. Non una vera easter egg rispetto a Yar's revenge, visto che il manuale ne accenna e scoprirlo è indispensabile per raggiungere il punteggio massimo, ma comunque gradito.

Portare a termine l'avventura non è affatto semplice, sia per la difficoltà derivata dalla parte di azione del gioco, sia per puzzle non banalissimi. Per recuperare gli oggetti che ci occorrono, capiterà spesso di dover cercare più volte negli stessi posti o farlo in determinati momenti e una volta trovati, indovinarne l'uso corretto non è sempre facile. Non dare nulla per scontato e saper aspettare è una delle chiavi del successo e saranno necessari un buon numero di tentativi e di partite per raggiungerlo. Il manuale a corredo della cartuccia è generoso nel fornire indicazioni ed indizi per avanzare nelle prime fasi dell'avventura dando addirittura istruzioni per scoprire i primi segreti, pur avvertendo che in questo modo ci si priva di parte del divertimento.

Conclusioni

Nonostante i suoi limiti e la sua età, I predatori dell'arca perduta è un gran bel gioco, ora come allora. Atari ha inventato le avventure dinamiche con Adventure e le ha migliorate con questo titolo introducendo elementi di azione e combattimento, anche se in forma primitiva, dando il via ad un genere apprezzato ancora oggi dal grande pubblico. Non è troppo azzardato (purtroppo) dire che i titoli da Tomb Raider in poi non hanno inventato nulla di nuovo, se non aggiungere la terza dimensione. A differenza delle moderne tendenze, questa avventura è completamente non lineare; non vi basterà semplicemente andare avanti e risolvere qualche puzzle più o meno semplice che vi viene piazzato davanti e di cui avrete probabilmente già tutti gli elementi per risolverlo. Qui, nei limiti di una console di 40 anni fa, siete liberi di girare per tutto il mondo di gioco ed interagire con esso (non credevate che gli open world fossero un'invenzione moderna, vero?), imparare a viverci dentro e scoprirne i suoi segreti prima di vedere come va a finire. Anche con l'aiuto dei suggerimenti dati dal manuale di gioco, saranno necessari giorni per avere abbastanza dimestichezza da sopravvivere alle insidie del Cairo e trovare la soluzione. Conclusioni? Fatevi un regalo e giocatelo, sull'hardware originale o perché no, anche su un emulatore.

di Giorgio Balestrieri

Yie Ar Kung Fu

Konami - Anno 1985 - Piattaforma MSX

Lang

Fanciulla tanto graziosa quanto letale, colpisce senza errore a distanza con i suoi shuriken, ma ancora più pericolosa nelle sue tecniche di kung-fu, in grado di tenere a debita distanza il proprio avversario, difficile avvicinarsi.

Tao

Personaggio alquanto inquietante, obeso, dagli improbabili calzoni rosa, ma capace di sputarci contro fiamme letali da evitare assolutamente, mentre se ci si avvicina troppo non ci pensa due volte a dare pugni e calci per difendersi.

Wang

Un grosso e pericoloso energumeno che pratica l'antica disciplina del Bo, intenzionato con il suo bastone a darcele di santa ragione. Nonostante la sua mole è piuttosto agile, schiva attacchi e colpisce a ripetizione sia con la sua arma che a suon di calci.

Sin dalla prima diffusione commerciale dei sistemi MSX, quello che riguardava il settore ludico, non aveva ancora basi solide e concrete, al punto che data la stretta somiglianza (o meglio eguaglianza) con sistemi come Colecovision e Sega SG1000, ha fatto sì che i primi videogiochi per lo standard di Nishi, fossero porting provenienti proprio dalle suddette console.

Tuttavia lo standard MSX era destinato, almeno in Giappone, a diventare uno dei due sistemi di riferimento dell'informatica domestica e commerciale, insieme ai PC NEC 88/9801.

Tra i primi colossi coinvolti nello sviluppo games per MSX, si distinse senza ombra di dubbio Konami, con un palinsesto di successi ed esclusive MSX che hanno dettato decisamente storia.

Uno tra i classici più ricordati e rinomati tra i videogiochi MSX targati Konami è Yie Ar Kung Fu, sviluppato direttamente dalla casa nipponica nel 1985, come versione ad hoc (come per quella NES di Nintendo), più che conversione, dell'omonimo titolo arcade da sala.

La decisione di recensire proprio questo titolo sta nel fatto che lo stesso mette in risalto alcune caratteristiche hardware MSX, con risultati davvero ottimi.

Iniziamo un passo per volta descrivendo in primis le differenze sostanziali tra questa versione MSX e quella arcade; innanzitutto si può dire che dall'aspetto grafico complessivo, si potrebbe tranquillamente dire di avere a che fare con due titoli completamente distinti, avendo solo il titolo in comune.

Il protagonista si chiama Lee, anziché Oolong della versione arcade, mentre gli avversari da affrontare sono cinque anziché dieci, più un livello bonus che si alterna dopo il secondo e quinto livello, non è previsto un boss finale, ma nuovamente la stessa sequenza di avversari, con un livello di difficoltà maggiore.

Il sistema è a barra di energia, quindi non è possibile eliminare il proprio avversario con un solo colpo, ma solo quando questa è ridotta a zero. Il numero limitato di avversari non deve però scoraggiare in quanto ognuno offre un ottimo livello di sfida e poco prevedibile con caratteristiche diverse tra loro.

Essi sono così elencati:

1. **Wang**: un grosso e pericoloso energumeno che pratica l'antica disciplina del Bo, intenzionato con il suo bastone a darcele di santa ragione. Nonostante la sua mole è piuttosto agile, schiva attacchi e colpisce a ripetizione sia con la sua arma che a suon di calci.

2. **Tao**: personaggio alquanto inquietante, obeso, dagli improbabili calzoni rosa, ma capace di sputarci contro fiamme letali da evitare assolutamente, mentre se ci si avvicina troppo non ci pensa due volte a dare pugni e calci per difendersi.

3. **Chen**: muscoloso, dalla barba cattiva, sadico e altrettanto pericoloso nell'armeggiare e roteare la sua micidiale catena che lancia con grande abilità e precisione, costringendo il giocatore a destreggiarsi in abbassamenti e salti per evitare di subire danni seri. E' in grado di sferrare anche potenti calci in caso ci si avvicini troppo.

4. **Lang**: fanciulla tanto graziosa quanto letale, colpisce senza errore a distanza con i suoi shuriken, ma ancora più pericolosa nelle sue tecniche di kung-fu, in grado di tenere a debita distanza il proprio avversario, difficile avvicinarsi.

5. **Wu**: praticamente un bisonte volante, il più pericoloso, una mole da armadio a due ante, colpi veloci e taglienti, ma quel che è peggio è il suo improvviso lanciarsi in volo verso il proprio avversario, con grande rischio di essere presi in pieno.

Di contro il nostro Lee, può contare sull'antica ed nobile arte marziale del Kung Fu per un efficace contrattacco. Le tecniche a disposizione prevedono pugno al corpo, calcio alto, calcio alla gamba, spazzata e calcio volante, oltre che salto e accovacciarsi per schivare gli attacchi.

Chen

Muscoloso, dalla barba cattiva, sadico e altrettanto pericoloso nell'armeggiare e roteare la sua micidiale catena che lancia con grande abilità e precisione, costringendo il giocatore a destreggiarsi in abbassamenti e salti per evitare di subire danni seri.

Wu

Praticamente un bisonte volante, il più pericoloso, una mole da armadio a due ante, colpi veloci e taglienti, ma quel che è peggio è il suo improvviso lanciarsi in volo verso il proprio avversario, con grande rischio di essere presi in pieno.

GIUDIZIO SUL GIOCO

GIOCABILITA'

95%

E' tipico di Konami scrivere i propri titoli per l'home gaming mantenendo una gameplay al pari di un coin-op nonostante le limitazioni grafiche del caso.

LONGEVITA'

75%

Nonostante la versione MSX non abbia riscontro con quella del coin-op se non per il nome, Yie Ar Kung Fu, offre un'ampia fetta di divertimento, anche se nonostante il livello degli avversari tende ad aumentare, una volta comprese le loro tecniche, il gioco può diventare un po' ripetitivo.

Il livello bonus che si alterna dopo il secondo e quinto avversario consiste nel colpire oggetti volanti come vasi, mattoni, piatti ed altro al fine di ottenere un super bonus finale, evitarli non aiuterebbe col punteggio, mentre se si colpiscono tutti gli oggetti il computer assegnerebbe un super bonus finale, ma se si viene colpiti il livello bonus si chiude senza però perdere una vita.

Come già detto, una volta sconfitti tutti e cinque gli avversari, il gioco riprende con un livello di difficoltà superiore sia la velocità di movimento dei vari guerrieri che della loro ferocia.

Tutto il gioco si svolge all'interno di un tempio shaolin che cambia colore ogni volta che si completano tutti e cinque gli stage.

La grafica è apparentemente semplice, potevano essere usati più dettagli e color, ma i personaggi sono davvero caratteristici, persino nelle espressioni del viso, cosa che nell'arcade non si nota. I controlli sono ottimi e immediati, non vi sono latenze di alcun tipo, sembra di giocare a livelli di un gioco da sala, mentre la colonna sonora è unica per ogni livello, carina e a tema, ma dopo un po' diventa ripetitiva.

La scelta di recensire questo gioco sta nel fatto che per la sua realizzazione vengono usate contemporaneamente due tecniche grafiche completamente diverse tra loro: lo sprite principale, ovvero Lee è composto da più oggetti, dato che la matrice massima di ogni sprite hardware su MSX1 è di 16x16 pixel e con l'handicap di avere un solo colore, per questo motivo il protagonista viene "costruito" da diversi sprite in modo da vere una figura più grande e più colorata. Tuttavia la VDP dell'MSX1 (il TMS9918a montato anche su Colecovision, Sega SG1000, TI99 e Memotech) pone un altro limite sulla gestione degli sprite hardware, ossia quella di non poterne visualizzare più di quattro sulla stella linea, con conseguenza che il successivo dal quinto in poi, con priorità di visualizzazione più bassa viene oscurato se viene allineato agli altri quattro.

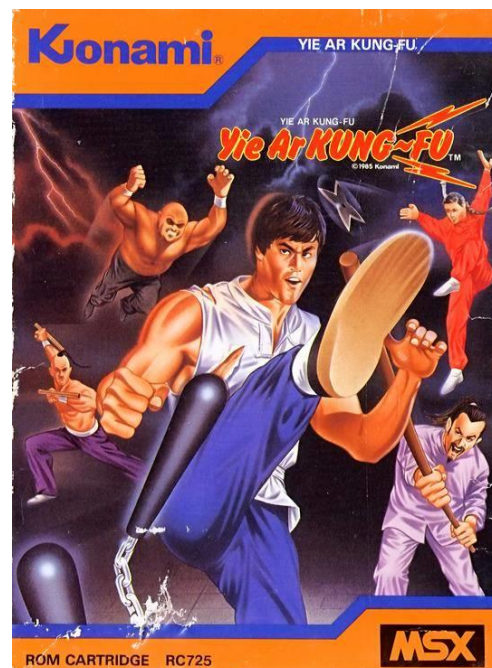
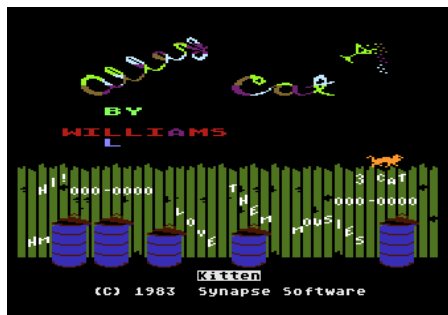
A questo punto qualcuno potrebbe domandarsi, "ma allora com'è stato possibile realizzare questo videogioco se sia il protagonista che l'avversario di turno occupano orizzontalmente ben piu di quattro sprite hardware?". La risposta è semplice: grazie alle tile ridefinibili (charset) con le quali vengono realizzati gli sprite software che rappresentano gli avversari. In pratica e molto sommariamente, ogni avversario è costruito attraverso diversi caratteri ridefiniti e animati con estrema velocità. Tutto questo grazie ad una VRam dedicata all'interno dell'MSX, completamente mappata ed esterna alla Ram di sistema, nella quale vengono riservati spazi appositi per la ridefinizione del charset e la ricollocazione di esso sullo schermo in modo dinamico. Il gioco infatti sembra dall'aspetto semplice ma è uno di quei titoli che sfrutta ottimamente la potenza e l'efficienza di calcolo dello Z80a montato come CPU, ma ancor di più un BUS dati dedicato appositamente alla VRam e alla VDP, oltre a quello esistente per la Ram di sistema.

Un titolo che mostra come Konami avesse ben inteso come lavorare in modo redditizio su MSX e quanto la macchina stessa offrisse tutte le possibilità di ovviare ai suoi stessi limiti in modo efficiente.

Il gioco venne pubblicato e distribuito per la prima volta in formato cartuccia Rom da16Kbyte, ma nel 1988, venne ripresentato in formato Floppy all'interno della Konami Games Collection n.1, unitamente ad altri classici di successo, di cui parleremo in seguito, con la particolarità di poter sfruttare, se posseduto, l'aggiunta sonora SCC, un formato audio custom MSX da parte di Konami/Yamaha, che aumenta in modo esponenziale le capacità sonore del sistema.

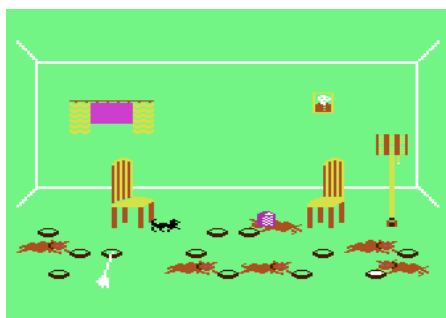
Infine è bene precisare che nonostante nel resto del mondo la versione di Yie Ar Kung Fu più conosciuta sia quella arcade da sala, in Giappone viene identificato nelle versioni MSX/Nes, dove gadget e merchandising fa riferimento unicamente a queste versioni.

di Francesco Gekido Ken Ugga

Alley Cat

Synapse Software - Anno 1983 - Piattaforma Atari 8-bit

La title screen

Backstreet's cat alright

Il vicolo-hub

Il punto di partenza per tutte le scorribande del "gattaccio".

La fase "stealth"

Se non sarete più che silenziosi nel soffiare il cibo dalle ciotole... cagnara assicurata.

La "boccia dei pesci"?

Vero che gatti generalmente odiano esssere bagnati... ma altrettanto vero che amano il pesce...

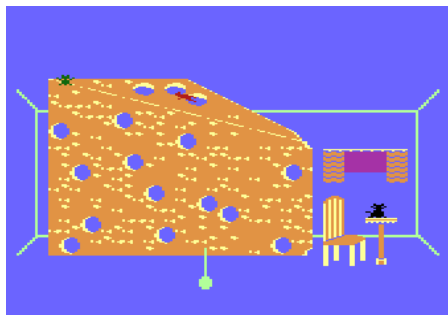
Il felino protagonista di questo brillante multi-game action firmato dall'estroso quanto umanamente sfortunato designer e programmatore Bill Williams (1960-1998) non è certamente il classico pigro e ben nutrito gatto da appartamento. Si tratta viceversa di un tipico randagio dedito a tutte quelle attività moleste che lo rendono sgradito alle massaie e sempre a rischio di incontri ravvicinati del tipo canino. Per certi versi, dunque, si potrebbe definire Alley Cat, letteralmente "Gatto da Vicolo", una "simulazione di gattaccio", fermo restando che quest'ultimo, per quanto bisbetico, non è affatto insensibile al richiamo dell'amore...

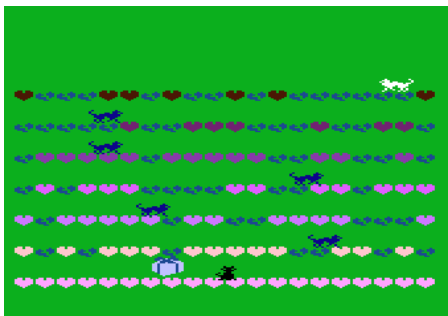
Questo flip-screen action sviluppato a partire da un concept di John Harris e pubblicato per Atari 8-bit dall'etichetta americana Synapse Software nel 1983 si articola in otto diverse videate, di cui sette raggiungibili da una principale, il vicolo per l'appunto, che funge dunque da "hub". La partita inizia proprio da questo scorcio di sobborgo popolare, dove il nostro pestifero gatto nero di nome Freddy deve subito saltare su uno dei bidoni e poi sulla staccionata, evitando così di azzuffarsi con un cane "di ronda" nella strada, cosa che gli costerebbe la perdita di una vita. A questo punto, sfruttando i fili scorrevoli dei panni stesi (e catturando nel frattempo qualche topolino acrobata), il felino dovrà balzare in una delle dodici finestre del palazzo, periodicamente aperte dalle già menzionate massaie "gattofobe" per bersagliarlo con lanci di scarpe e altri oggetti.

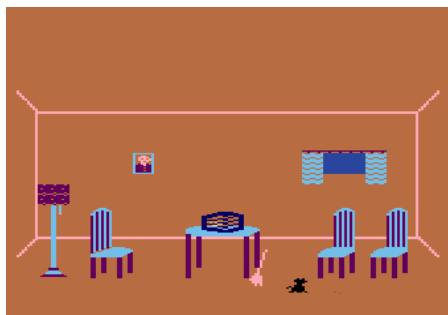
Una volta entrato si troverà ad affrontare varie sfide che prendono le mosse da situazioni plausibili e le rendono con accattivanti tocchi vagamente onirico-visionari se non proprio surrealistici, tanto da sembrare quasi sogni agitati del micio stesso a seguito di una scorpacciata di pesce particolarmente indigesta. Al di là di quest'ultimo aspetto, peraltro coerente con la spumeggiante estrosità creativa di Bill Williams (solo l'anno prima aveva creato il quanto mai peculiare Necromancer e in seguito firmerà Mind Walker e Pioneer Plague), i contesti proposti sono appunto chiaramente ispirati a controparti reali: far cadere da un tavolo la gabbia dell'uccellino per poi catturarlo prima che voli via, raggiungere la boccia dei pesci posta su un altro tavolo per farne un pasto prelibato e rubare il cibo dalle ciotole dei cani senza svegliarli (fase "stealth"). Ovviamente per portare a termine queste "missioni" il felino dovrà vedersela con avversari e minacce: la scopa intenta a mantenere puliti i pavimenti (e scacciare eventuali intrusi a quattro zampe) e il sonno particolarmente leggero dei ringhiosi custodi delle ciotole stesse.

Il tocco visionario dell'autore è espresso ad esempio nella rappresentazione ingigantita della fish bowl: una vera e propria vasca pullulante di pesci da acchiappare e anguille elettriche da evitare, cercando al contempo di non restare a corto di ossigeno durante le reiterate immersioni. Un gigantismo non dissimile lo si può poi notare in un'altra sfida: una stanza dominata da un enorme pezzo di formaggio con i buchi. Qui il nostro Freddy dovrà catturare alcuni topi inseguendoli nei buchi di questo "maxi-cacio". Un'ulteriore sfida vede inoltre il felino alle prese con una libreria, con alcune piante da raccogliere poste in cima ad essa e con vari ragni extralarge da evitare durante l'operazione.

Interessante notare due tocchi di classe che accentuano il ritmo di gioco e rendono più intrigante il gameplay: il balzo attraverso le finestre di Freddy avviene in entrata come in uscita e la succitata scopa può essere "distratta" dalle impronte lasciate dal micio stesso. In altri termini un salto fuori misura da un tavolo può tradursi in una rovinosa "ricaduta" nel cortile (vicolo) attraverso la finestra e qualche "zampettamento strategico" sul pavimento di certe stanze costituisce un ottimo diversivo per tenere occupata la ramazza.

Manca però all'appello l'ultima sfida: quella del cuore. Freddy, infatti, vuole raggiungere a tutti i costi il suo amore, Felicia, e per farlo dovrà ottenere un buon punteggio nelle varie prove e far così apparire la gattina dei suoi desideri sul davanzale di una delle finestre. Se il nostro micio balzerà dunque nella finestra con il giusto tempismo affronterà la sfida più importante, ovvero quella che potrebbe coronare il suo sogno d'amore... e fargli guadagnare una vita extra. In questa schermata, non a caso tra tutte la più onirico-visionaria, Freddy dovrà raggiungere e baciare Felicia balzando su una serie di piattaforme composte da cuori, fronteggiando una serie di gatti rivali pronti a soffiargli contro per respingerlo (possono essere però distolti temporaneamente con...

Il maxi-cacio

Niente di più normale di un gatto che insegue dei topi nei buchi di una forma di formaggio, no?

La prova del cuore

Riuscirà il nostro Romeo felino a raggiungere la sua Giulietta dal mantello bianco?

La gabbia dell'uccellino

"Mi è semblato di vedele un gatto" (cit.)

GIUDIZIO SUL GIOCO

GIOCABILITA'

90%

Immediato come un coin-op e inizialmente tutt'altro che difficile. Il nodo della difficoltà tuttavia non tarderà a stringersi e il ritmo di gioco, già serrato in partenza, si farà sempre più incalzante.

LONGEVITA'

80%

Le sfide proposte al nostro amico felino offrono più varietà di quanto non possa sembrare di primo acchito e risultano ancora più godibili poiché di fatto, nel loro rapido susseguirsi, non spezzano la continuità di gioco.

un pacco regalo) e stando soprattutto attento alle frecce scagliate da una processione di Cupidi. Alcuni di questi dardi, infatti, cancelleranno progressivamente tutte le piattaforme, fino a far tristemente precipitare l'innamoratissimo felino.

Benché sia stato sviluppato solo su sistemi domestici, vale a dire prima su home computer Atari 400/800/XL/XE (48 kB RAM) e poco dopo su PC (CGA), il divertente Alley Cat propone un ritmo di gioco abbastanza veloce e serrato da renderlo assimilabile sotto questo aspetto a un coin-op dei primissimi anni '80. Contribuisce poi a questa sua spiccata inclinazione arcade "vecchia scuola" la mancanza di una vera e propria fine. Ogni volta infatti che Freddy riesce a baciare Felicia ottiene una vita e incrementa la difficoltà, fino a un massimo di 30. Raggiunto eventualmente tale skill level il giocatore potrà vantarsi a ragion veduta di avere riflessi davvero felini, mentre la partita proseguirà senza elevare l'asticella della sfida "fino ad esaurimento vite".

Dal punto di vista grafico il titolo Synapse risulta assolutamente apprezzabile in rapporto alle specifiche degli Atari 8-bit e all'anno di rilascio. Gli sprite sono sì monocromatici ma non presentano rilevanti flickerii e convincono pienamente sul fronte della fluidità e delle animazioni, mentre l'utilizzo dei colori beneficia in alcune schermate dei "DLI tricks", vale a dire di Display List Interrupts atti ad incrementare il numero di sfumature a video, sia pur con stringenti limitazioni nella relativa giustapposizione e una sorta di "viraggio psichedelico" del look complessivo.

Particolarmente valido infine il lato sonoro, reso assolutamente peculiare dai buffissimi quanto sorprendentemente "elaborati" effetti sonori e musicalmente ben presentato nella title screen da una simpatica quanto bizzarra chiptune cover con accompagnamento di miagolii di "Alley Cat" (1963), popolare brano jazz di Bent Fabric.

di Alessio Bianchi

La libreria

A giudicare dai ragni il padrone di casa non deve essere un amante della lettura...

Boccia dei pesci che passione!

Il "Posso resistere a tutto tranne che alle tentazioni" vale anche per i gatti.

Intervista ad **Antonio Savona**

PROGRAMMATORE DI **PLANET GOLF** PER COMMODORE 64

Non fa notizia che Antonio Savona, recentemente salito alla ribalta per aver programmato l'incredibile Planet Golf – uno dei migliori titoli per Commodore 64 usciti lo scorso anno – sia un appassionatissimo di videogiochi. Ma sicuramente è molto più interessante sapere come sia un videogiocatore di vecchia guardia da sempre ultrafan del Commodore 64 e come si sia arrivati alla realizzazione di un gioco, a dir poco stupefacente, sull'amatissimo 8-bit nel 2017.

È una passione coltivata fin dall'adolescenza che poi Savona, oggi 44enne, ha trasformato in lavoro visto che 8 anni fa si è trasferito a Londra per entrare in Microsoft come Data Scientist collaborando anche alla realizzazione del motore di ricerca Bing. Adesso è sempre impegnato in ambito informatico ma è passato ad un'altra compagnia.

"Sono sempre stato un collezionista di hardware – spiega Antonio Savona – ed un giorno ho tirato fuori un floppy disk di un gioco che stavo sviluppando quando avevo 15 anni. Lo ho provato ed è partito. A quel punto mi dissi che dovevo finirlo ma invece di riprenderlo ho iniziato a fare Po Snake".

E come è andata?

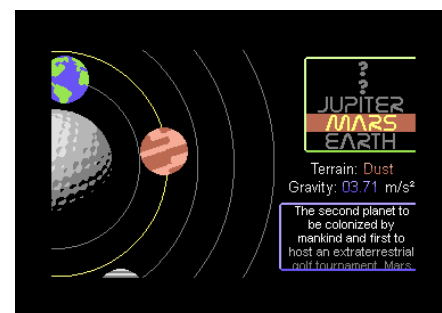
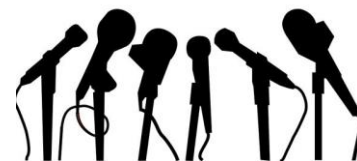
"È andata molto bene: ho pure vinto una competizione, la RGCD 16K Coding Competition che, come suggerisce il titolo, consiste nella realizzazione di un gioco con 16 kbyte di spazio. Il gioco vinse il contest ed il diritto di essere pubblicato commercialmente da un publisher".

Iniziamo realmente l'intervista con una domanda di rito: cosa ti ha spinto a realizzare un gioco per Commodore 64 nel 2017?

"Come detto è in realtà un ritorno alla programmazione dopo Po Snake. Volevo fermarmi li, poi ho provato a fare un gioco più complesso... ed effettivamente (ride) è venuto più complesso".

Come mai ha scelto di realizzare proprio un gioco di golf (mascherato da meccaniche puzzle)?

"Non conoscevo Stickman Golf che è un titolo simile al mio. Inoltre, originariamente, volevo fare un platform con una fisica fatta bene. Volevo creare una serie di algoritmi e routines con fisica realistica per i rimbalzi, la gravità ed utilizzarle per un platform perché su C64 queste librerie complesse non ci sono e le devi per forza fare da solo se vuoi un risultato apprezzabile. Quando ho sviluppato quello che possiamo chiamare un engine (ovviamente rudimentale) per la fisica, ho visto che non potevo farci girare troppi oggetti. L'hardware del Commodore 64 è quello che è. E da li è venuta l'idea di una pallina da golf. Questa mi ha fatto venire un'altra idea: cambiare la gravità e rendere più vario il gioco mettendo più pianeti e non limitandosi solo al nostro. Infatti, in Planet Golf, troviamo 5 pianeti: la Terra, Marte, Giove e due pianeti che possono essere sbloccati tramite achievements (cosa rara su un C64, ndr), Nettuno e Kepler, un esopianeta. Ovviamente la gravità diversa offre spunti molteplici per il gameplay ed ogni pianeta, inoltre, presenta diversi ostacoli oltre alla già sopracitata gravità differente. Troviamo, infatti, anche diverse tipologie di terreno che influiscono sul movimento della pallina. Su Nettuno, ad esempio troveremo un attrito scarso mentre come ostacoli avremo di fronte dei laser. È inoltre possibile incontrare alcuni elementi di contorno. Sulla terra i piccioni, su altri pianeti gli alieni, e così via per sbloccare gli achievements necessari ad andare su Nettuno e Kepler".

Interessante lo spunto che vede protagonista il golfista Alan Shepard che

gioca sulla luna. Da lì abbiamo visto è partito tutto il resto. Come è nata la scintilla?

"Sono appassionato di esplorazioni spaziali. Il mio idolo è, appunto, Alan Shepard, una persona incredibile. Era un vero eroe. Era uno dei Mercury Seven che ha fatto partire il programma spaziale. Da pilota ha studiato autonomamente e si è costruito un bagaglio tecnologico tale da apportare importanti innovazioni al programma che ha consentito all'uomo di andare sulla Luna. È stato il primo americano nello spazio (qualche tempo dopo il russo Yuri Gagarin, ndr)... era appassionato di golf. Ed il gioco ha tante citazioni su di lui come ad esempio la frase Miles, Miles and Miles che esclamò quando colpì per la prima volta la pallina sulla Luna descrivendone la gittata del suo tiro. Nel gioco troveremo altre sue citazioni".

Ti saresti aspettato questa reazione sul web e gli ottimi risultati ottenuti finora?

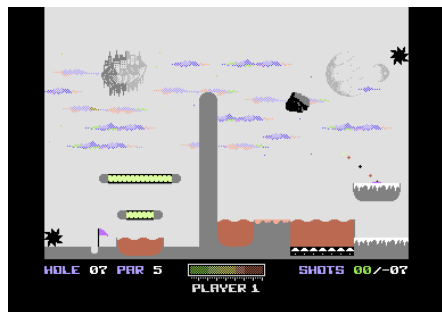
"Ovviamente no. Il mio gioco precedente ha fatto sold-out subito. Ma di Planet Golf non me l'aspettavo. Su disco ha venduto tantissimo già nella prima settimana (con i proventi miei che devolvo in beneficenza). Sono ovviamente contento. Mi ha sorpreso comunque perché benché il titolo sia molto fatto bene anche per merito di chi ha collaborato con me, non pensavo che un gioco di golf tirasse tanto. Poi ho avuto la fortuna di conoscere Oliver Frey che ha realizzato la copertina di Planet Golf e che è

lo storico autore delle cover di Zzapp (pietra miliare in ambito editoriale essendo una delle prime riviste del settore dedicata agli 8 bit che venne venduta in Italia fino ai primi anni '90)".

Da quali giochi hai tratto ispirazione per Planet Golf?

"Da nessuno. Ho tratto ispirazione da Alan Shepard che ha giocato a golf sulla Luna. Anzi, più che ispirazione l'ho proprio copiato miseramente".

Quali sono i tuoi giochi per C64 preferiti?

"Sono tantissimi ma ne cito solo tre. Magari non tutti il conosceranno. In primis, Archon del 1983 che è uno strategico. Si gioca su una scacchiera che ospita pezzi raffiguranti creature mitologiche che quando si scontrano combattono. Un po' simile a Battle Chess ma qui il combattimento è gestito da una sessione arcade che invita il giocatore ad agire e non era già deciso come nel famoso gioco scacchistico. Questo aveva anche una copertina in stile vinile ed aveva un'altra particolarità: era stato sviluppato da una donna Anne Westfall una cosa non comune all'epoca. Poi The Last Ninja per la realizzazione tecnica. Ho anche incontrato John Twiddy e Ben Daglish (il primo è l'autore del gioco, l'altro ha realizzato alcuni pezzi delle colonne sonore, veri e propri capolavori, ndr). Turrican, infine, un gioco fuori-scala per l'epoca. Una realizzazione miracolosa".

In quanto tempo hai realizzato il progetto e quali tools hai utilizzato?

"Planet Golf ha richiesto un anno perché ho dovuto ritagliarmi lo spazio necessario per lavorarci su. Al netto, diciamo così, due-tre mesi di sviluppo. Ci ho passato molte serate. Ho lavorato in Assembly ed usato ottimi assemblatori come il Kick Assembler. Normalmente si utilizzerebbero tool grafici e sonori preesistenti ma in questo caso, visto il concept inedito, ho realizzato editor e tools da me. Ho fatto un encoder audio per comprimere le musiche e gli effetti sonori e l'editor dei livelli. Il gioco complessivamente occupa 360 kbyte. La sola intro, in full motion video, occupa 150 kbyte e ci sono 2 minuti di video ed audio digitalizzati".

Hai *lottato* con l'hardware o tutto è venuto semplice?

"Parecchio. Ho dovuto scrivere da zero i programmi per i calcoli. Il solo rimbalzo se calcolato bene, richiede operazioni in virgola mobile complesse. Il C64 non ha i mezzi per realizzarle e quindi bisogna implementare questa caratteristica. Ho dovuto fare questo per eludere la barriera tecnica della fisica. Anche il video su C64 non è stata propriamente una passeggiata di salute".

Ultima domanda, anch'essa di rito, che programmi hai per il futuro? Il sequel, una versione Amiga, o un nuovo gioco?

"Ci sono un paio di progetti che stanno prendendo forma. Il primo lo avevo iniziato precedentemente al progetto di Planet Golf e si chiama Guy in a Vest. Sempre per C64 ovviamente. Si tratta di un run and gun stile classico perché vorrei fare una cosa un po' meno ricercata con un gameplay nudo e crudo. Mentre il publisher mi ha chiesto di fare un'espansione per Planet Golf. Sono contento ma mi piacerebbe fare il sequel aggiungendo elementi di gameplay diversi e sfruttando i feedback dei giocatori. Non, quindi, una semplice espansione. Se tutto andrà bene si lavorerà nel 2019. Quest'anno lo vorrei dedicare a Guy in Vest. Tornando al nuovo Planet Golf, vorrei fare un gioco con caratteristiche diverse come ad esempio percorsi più lunghi e quindi livelli dotati di scrolling. Vorrei chiamarlo World Class Planet Golf in omaggio alla vecchissima, gloriosa e celebre saga Leaderboard".

Grazie per il tuo tempo.

"Grazie a voi per l'opportunità. È un piacere. Devo dire che ho ricevuto diverse richieste per fanzine straniere e nessuna richiesta italiana. Strano a dirsi ma è così".

Come dire... nemo propheta in patria...

di Edoardo Ullo

Eventi: Firenze Vintage Bit 2017

di Leonardo Vettori

Anche quest'anno è arrivata puntualmente la nona edizione del "Firenze Vintage Bit", la manifestazione a tema retrocomputing organizzata dall'associazione omonima nell'antico Spedale quattrocentesco di Sant'Antonio di Lastra a Signa, a pochi chilometri da Firenze.

L'evento si è svolto Domenica 26 Novembre ed ha richiamato espositori da ogni parte d'Italia, registrando una grande partecipazione di pubblico, che ogni anno cresce sempre di più.

Firenze Vintage Bit nasce grazie all'impegno di Walter Pugi e Maurizio Morandi e dal 2009 si pone come una manifestazione ad ingresso libero e totalmente priva di aspetti commerciali.

Lo scopo è far rivivere le macchine, facendole provare liberamente al pubblico. Interessante notare che, oltre agli appassionati del settore ed ai nostalgici, si registra una sempre più alta partecipazione di giovani, che rimangono spesso incuriositi da un mondo

informatico oggi tanto lontano quanto affascinante.

Altra caratteristica unica della manifestazione è la location; l'antico Spedale di Sant'Antonio, che viene ogni anno messo gratuitamente a disposizione dal Comune di Lastra a Signa, crea un particolare contrasto con la manifestazione di retrocomputer, dandole un fascino senza dubbio particolare.

Come i più informati già sapranno, ogni edizione prevede un tema particolare che fa da base per conferenze ed esposizioni. Quest'anno è stato doveroso festeggiare i primi trent'anni della storica Amiga 500, una macchina semplicemente immortale, amata ancora oggi da una nutrita schiera di appassionati. Ci sono state ben due conferenze dedicate al computer di casa Commodore; la prima, tenuta la mattina, si è focalizzata sulla genesi dell'hardware, l'impatto al momento dell'uscita, le caratteristiche che lo rendono unico e il lascito che ha avuto. A tenere viva l'attenzione degli spettatori ci ha pensato Leonardo Vettori, vicepresidente di Associazione Firenze Vintage Bit Onlus, coadiuvato dagli interventi tecnici di Stefania Calcagno ed Alberto Peri.

Nel pomeriggio invece si è tenuta la seconda conferenza, curata dalla programmatrice Stefania Calcagno, che ha raccontato l'evoluzione della macchina Commodore ed il mondo della Demoscene, di cui la relatrice stessa fa parte. Stefania ha anche portato per l'occasione l'unico prototipo funzionante di Amiga Walker, il cosiddetto successore dell'Amiga 1200, che le è stato donato da Petro Tyschtschenko direttore della logistica di Commodore International Ltd.

Ovviamente l'esposizione non era certo riservata al solo mondo Amiga, anzi; quest'anno non sono mancati tutti gli altri componenti della famiglia Commodore: Vic 20, C64, C128, C116, C16, Plus/4, CDTV e CD32. Molto gradita inoltre è stata la presenza del programmatore Luca Carrafiello, che ha portato diversi nuovi giochi in sviluppo per C16 e Plus/4; il pubblico di ogni età ha particolarmente apprezzato, affollando la famiglia 264 come mai prima d'ora.

Oltre a questo comunque non sono mancati i computer Sinclair, Atari, IBM, Olivetti, Acorn, Texas Instruments ed MSX, accompagnati da monitor, tastiere musicali, disk drive, tavolette grafiche, Joystick, digitalizzatori video e tanto altro. Impressionante è stata anche la grande esposizione di computer di casa Apple di Robert Swiderski, che ha colpito tutti i presenti.

Anche il mondo console è stato preso in considerazione, grazie all'esposizione di sistemi come: GCE Vectrex, Sega Megadrive, Philips CD-i, Atari Jaguar e la prima Playstation.

Novità dell'anno è stata l'esposizione di formati video "particolari" come il Laserdisc e il CED, che hanno riscosso un buon successo tra il pubblico. La manifestazione è stata un grande successo ed i lavori per l'edizione 2018 sono già iniziati.

Vi diamo quindi appuntamento fin d'ora per **l'ultima domenica di Novembre 2018 per la decima edizione del Firenze Vintage Bit**, dove i computer dei "vecchi tempi" rivivono e si mostrano più vivi ed in buona salute che mai. Per ulteriori informazioni visitate il sito www.retrocomputer.org.

La posta di RetroMagazine!

Da questo numero diamo il via ad una nuova rubrica: **la posta di RetroMagazine!** Questo spazio e' per voi, per le vostre domande, le discussioni e perche' no anche per le critiche e le correzioni... Scriveteci all'indirizzo: RetroMagazine.redazione@gmail.com

Scusate se vi faccio perdere tempo, ma visto che scaricando e leggendo RetroMagazine (complimenti!!!) mi è venuta voglia di recuperare il c64 in soffitta, mi permetto di scrivervi per una domanda sicuramente banale ma che mi incuriosisce parecchio. Visto che da bambino (ho ricevuto il c64 a Natale '84, avevo 10 anni) ci giocavo e basta, in questi giorni mi son messo a leggere il manuale e a provare i programmi; a pag. 65 del manuale c'è un programma per generare una pallina rimbalzante sullo schermo:

ALTRO SULLE PALLINE RIMBALZANTI

Qui c'è un programma revisionato per la pallina rimbalzante che stampa direttamente sullo schermo con i POKE anzichè usando i comandi del cursore all'interno delle istruzioni PRINT. Come si vedrà dopo aver eseguito il programma, la flessibilità è maggiore che non quella precedente e consente di programmare un'animazione molto più sofisticata.

```
NEW

10 PRINT "{CLR/HOME}"
20 POKE 53280,7 : POKE 53281,13
30 X = 1 : Y = 1
40 DX = 1 : DY = 1
50 POKE 1024 + X + 40*Y,81
```

65

```
60 FOR T = 1 TO 10 : NEXT
70 POKE 1024 + X + 40*Y,32
80 X = X + DX
90 IF X = 0 OR X = 39 THEN DX = -DX
100 Y = Y + DY
110 IF Y = 0 OR Y = 24 THEN DY = -DY
120 GOTO 50
```

se imposto le variabili DX e DY al valore 2 invece che 1, il C64 va in bomba e restituisce listati infiniti invece di poche righe di programma. Ho pensato che il glorioso amico d'infanzia fosse andato, quindi ho installato il Vice sul mio Mac e ho ribattuto il programma: idem. La buona notizia quindi è che il C64 dovrebbe essere sano. Credo. Grazie per una risposta, altrimenti grazie lo stesso perché la vs pubblicazione è fantastica. Ciao! - **Andrea F**.

Risponde Marco Pistorio. Ciao Andrea! Il comportamento "anomalo" del C64 quando imposti DX e DY al valore 2 è dato dal fatto che il programma BASIC che hai digitato non riesce a "capire" quando deve invertire il movimento della pallina, a causa di come è effettuato tale controllo (vedi gli IF a riga 90 e 110). Così succede che il codice della pallina (81) ed il codice dello spazio vuoto per cancellarla (32) vengono scritti in aree di memoria non più dedicate allo schermo (locazioni da 1024 a 2023), sporcando ad esempio l'area destinata a contenere i programmi BASIC che inizia dalla locazione 2049. Ecco spiegato anche il motivo per cui il listato diventa "strano", come hai giustamente notato. Come risolvere il problema? E' presto detto. Basta far sì che il valore di X e di Y non vadano mai in negativo e non superino mai il valore 39 per quanto concerne X e 24 per la Y. In altre parole, basta aggiungere sotto la riga 80 le seguenti righe:

```
81 IF X<0 THEN X=0
82 IF X>39 THEN X=39
```

e sotto la riga 100 le seguenti righe:

```
101 IF Y<0 THEN Y=0
102 IF Y>24 THEN Y=24
```

In questo modo il controllo effettuato dagli if a riga 90 e 110 sarà più efficace e potrai impostare DX e DY a 2, oppure a 3 e così via. Spero che ti sia tutto chiaro.

Ciao a tutti, come potete immaginare sono un appassionato cultore del Retro Computing in tutte le sua sfaccettature, anche se purtroppo non ho mai avuto un approccio col mondo della programmazione che mi ha sempre affascinato. Ho iniziato con un C64 passando per Amiga e finendo, ahimè, al mondo dei PC. Mi sono sempre chiesto se Commodore fosse ancora una realtà, cos'avrebbe potuto fare e dove saremmo oggi con i computer rivoluzionari creati tra la fine degli '80 e l'inizio dei '90. Per ora mi limito a collezionare macchine e periferiche Commodore, che mi piace mostrare con orgoglio ai miei amici, essendo tutte perfettamente funzionanti e in condizioni estetiche più che dignitose. In tutto questo s'inserisce "RetroMagazine", che al pari di altre produzioni a tema, sia cartacee che in formato elettronico, mi hanno piacevolmente accompagnato alla continua ricerca del sapere, nel fantastico mondo di quegli anni in cui ci si divertiva in maniera genuina con un computer. Sono rimasto molto colpito dagli articoli sulla programmazione e v'invito a continuare su questo passo, proponendo da zero le tecniche e gli strumenti necessari a chi volesse avvicinarsi per la prima volta ad un mondo così complesso, almeno ai miei occhi inesperti! A tal proposito sarebbe bello creare una specie di corso a puntate, partendo proprio dall'inizio senza dare nulla per scontato, cercando di spiegare in maniera semplice anche i termini di uso comune per un programmatore (Es. Accumulatore, Interrupt per citare un paio di termini a caso). So bene che la cosa non è facile e richiede soprattutto parecchio tempo, così come sono consapevole che basterebbe una bella ricerca in rete per avere le risposte, ma credo che trovare tutto il necessario nello stesso posto, permettendo di seguire la lettura in maniera fluida e senza interruzioni sarebbe un valore aggiunto non da poco per il vostro Magazine. Anche le altre sezioni mi sono molto piaciute, come le recensioni dei giochi e le interviste. In generale tutta la rivista si legge con molto piacere e mi lascia con quella voglia di scoprire cosa ci sarà nel prossimo numero! Concludo facendovi ancora i complimenti per l'ottimo lavoro e mi auguro che possiate continuare a lungo, incrementando le collaborazioni e di conseguenza il contenuto del prodotto che si presenta già dai primi numeri in maniera molto professionale e interessante. – ***Ugo C***.

Risponde Francesco Fiorentini. Ciao Ugo, ci fa immensamente piacere il fatto che apprezzi il nostro lavoro ed allo stesso tempo ci riempie di orgoglio sapere che stiamo andando nella direzione giusta. Vogliamo inoltre confermarti che abbiamo in programma di scrivere articoli a puntate che introdurranno la programmazione BASIC ed Assembly, non solo per il C64 ma anche per altre piattaforme come lo ZX Spectrum, l'Atari, l'Amiga e l'MSX... Ovviamente stiamo cercando di introdurre questi argomenti gradualmente per non scoraggiare nessuno ed allo stesso tempo mantenere un giusto equilibrio tra tecnico e ludico. Le idee sono molte ma e' l'apprezzamento di voi lettori che ci spinge ogni mese a dedicare un po' del nostro tempo libero per curare la rivista.

Ringraziamenti ed una anticipazione

Prima di passare alle anticipazioni vogliamo ringraziare a nome dell'intera Redazione di RetroMagazine gli amministratori delle pagine Facebook, Siti e Blog che ogni mese ci aiutano condividendo il nostro lavoro con voi lettori:

Retrogame Machines Italia
Retrogame Machines Italia -La Pagina-
Retropie Facebook Italia
Generazione PlayStation One
Salagiochi 1980
Naonian Retrogaming Society
Commodore 64/128 italia
Commodore amiga italia
Retrocommodore
Commodore Amiga & 8 bit Italia
Commodore Basic,Assembly & Petscii
Retrogems
Ready64
Multi System Retro Italia
Programmazione Commodore e Amiga
Retrocomputer, Retroconsole & accessori, passione e amicizia
MSXItalia
Retrogaming Town
The Retrogames Machine
Another-RetroWorld
Tiggers
Spectravideo
Associazione Firenze Vintage Bit Onlus
MSX Home Computers
MSX Boixos Club
Vicoretrò

I bit-elloni
Re.BIT Magazine

Vogliamo inoltre rivolgere un ringraziamento speciale ad **Edi Musso** che si e' attivamente adoperato per diffondere la rivista. Grazie!!!

Quando alziamo gli occhi al cielo e li rivolgiamo alla Luna, ai pianeti e alle stelle non ci poniamo solo davanti alla vastità del cosmo ma stiamo guardando anche indietro nel tempo. Vediamo la Luna com'era un secondo fa, il Sole otto minuti e quattordici secondi fa, Alfa Centauri quattro anni fa, la Galassia di Andromeda centomila anni fa. E via, sempre più lontano e sempre più indietro nel tempo. Non c'è nulla di più "retro" che guardare alle stelle e, al tempo stesso, nulla di più legato al presente e proiettato nel futuro. È per questa ragione che abbiamo deciso di aprire **RetroSpace**, una nuova rubrica in Retromagazine che ci farà guardare al presente e al futuro ripercorrendo le tappe che hanno portato l'Uomo dalle caverne fino mettere piede sulla Luna, a visitare altri pianeti e a spingersi oltre i limiti contingenti della conoscenza e della tecnologia. Nel corso della nostra cavalcata ripercorreremo rapidamente la storia dell'astronautica e delle principali missioni spaziali attraverso le storie dei protagonisti, di donne e uomini straordinari che hanno permesso al genere umano di lasciare la superficie del pianeta e produrre avanzamenti in ogni campo dello scibile. Poi, avvicinandoci al cinquantenario dello sbarco sulla Luna, studieremo in dettaglio il computer di navigazione delle missioni Apollo e progetteremo **RetroMagazineSAT I**, il nostro pico-satellite artificiale nella speranza di continuare a sognare di essere destinati alle stelle.

ERRATA CORRIGE: Nel numero 3 abbiamo riportato in una didascalia "Speak& Spell, conosciuto in Italia come Sapientino…" in realta' era conosciuto in Italia come il Grillo Parlante. Volevamo vedere se stavate attenti!! Decisamente lo siete stati visto il numero di segnalazioni. ☺

Disclaimer

RetroMagazine
Anno 2 - Numero 4

Direttore Responsabile
Fiorentini Francesco

Immagine di copertina
Flavio Soldani

Febbraio 2018